Tvorba a adaptace lingvistické vrstvy pro systém rozpoznávání mluvené češtiny

Jan Kolorenč

Tvorba a adaptace lingvistické vrstvy pro systém rozpoznávání mluvené češtiny

Disertační práce

Disertant:	Jan Kolorenč
Studijní program:	2612V Elektronika a informatika
Studijní obor:	2612V045 Technická kybernetika
Pracoviště:	Laboratoř počítačového zpracování řeči, Ústav informačních technologií, Fakulta mechatroniky a mezioborových inženýrských studií, Technická univerzita v Liberci
Školitel:	Prof. Ing. Jan Nouza, CSc.

Rozsah práce:

Počet stran:	103
Počet obrázků:	20
Počet tabulek:	31
Počet příloh:	2

Prohlášení

Tuto práci jsem vypracoval samostatně s využitím uvedené literatury a na základě konzultací se svým školitelem.

V Liberci 20. února 2007 Jan Kolorenč

Poděkování

Děkuji rodičům za poskytnutí zázemí nutného pro vytvoření této práce. Dále děkuji všem, kteří mě podporovali, za jejich podnětné a konstruktivní připomínky, které pomohly zvýšit úroveň tohoto díla. Zejména děkuji Janu Nouzovi za vytvoření a zapůjčení automatických rozpoznávačů řeči, bez kterých by tato práce nemohla vzniknout, Jindřichu Žďánskému za distribuovanou implementaci rozpoznávačů, která umožnila provést množství experimentů v přijatelném čase a Jindře Drábkové.

Anotace

Tvorba lingvistické vrstvy pro systém rozpoznávání mluvené češtiny je v tomto díle chápána jako komplexní úloha skládající se z logicky navazujících kroků. Jednotlivé kroky využívají různorodé přístupy od využití hrubé výpočetní síly přes metody umělé inteligence, využití rad expertů až po různé heuristiky. Často dochází k fúzi těchto přístupů.

Nejprve jsou diskutovány otázky různých zdrojů textových dat a problémy při jejich využití. Jsou též uvedeny metody čištění textového korpusu a jejich vliv na úspěšnost rozpoznávaní.

V části o slovníku a fonetickém přepisu je diskutován vliv velikosti slovníku. Dále je uvedena metoda pro semiautomatické nalezení nových fonologických pravidel vylepšující automatickou fonetickou transkripci. Přidáním slovních párů do slovníku lze téměř bezpracně zlepšit úspěšnost rozpoznávaní. Tato metoda je uvedena na závěr části týkající se slovníku.

Velký slovník způsobuje problém při implementaci počítání jazykového modelu. Tento problém je vyřešen pro různé konfigurace počítačů v závislosti na preferenci malé spotřeby paměti nebo rychlosti výpočtu. Dosavadní programy pro výpočet jazykového modelu jsou výrazně zrychleny, čímž mohlo být uskutečněno mnoho experimentů.

Pro efektivní zvyšování úspěšnosti rozpoznávání je nutné co nepřesněji identifikovat a kvantifikovat chyby. Je proto zlepšena metoda vyhodnocování výsledků rozpoznávání.

Adaptace jazykového modelu je v literatuře velmi diskutovanou částí automatického rozpoznávání řeči. Úspěšnost adaptace závisí na mnoha faktorech. Proto je uvedena řada experimentů ukazujících vliv adaptace jazykových modelů na úspěšnost rozpoznávání rozpoznávačů vyvinutých v Laboratoři počítačového zpracování řeči Technické univerzity v Liberci. Tyto experimenty bylo možné provést též díky výraznému zvýšení rychlosti rozpoznávačů a programů pro vytváření jazykového modelu.

Na závěr je uvedena metoda automatické interpunkce zvyšující čitelnost výstupu rozpoznávače spojité řeči. Uvedená metoda je schopna odhadnout pozici interpunkce pouze na základě výstupu rozpoznávače oproti jiným metodám vyžadujícím též přítomnost akustického signálu.

Annotation

Development of a language model layer for an automatic speech recognition system is understood as a complex task. This task consists of many logically following steps. Approaches used in these steps range from computational brute force, artificial intelligence, experts' help to several heuristics. Combination of different approaches is often required.

The first task is to collect text data. Several sources and their specific advantages and problems are discussed in this work. Collected text data are called text corpus. This corpus has to be cleaned before it is used. Cleaning methods partly depend on data source. Effectiveness of common cleaning methods are evaluated with respect to recognition accuracy.

Next step is to create vocabulary and assign phonetic transcription to each word in the vocabulary. Semiautomatic approach for creation of new phonological rules is presented. These rules are used in the automatic phonetic transcription. Multi-words in the vocabulary easily increase recognition accuracy. This part also discuss influence of vocabulary size on recognition speed and accuracy.

Language model computation is problematic when large vocabulary is needed. The computation requires large amount of memory. This problem is solved for different requirements. The first approach maximally saves required memory, the second one maximizes computation speed. Current software for language model computation is significantly improved, so many experiments can be performed.

Effectiveness of speech recognition improvement depends on proper experiment evaluation. The better mistakes are identified the more effective recognizer's enhancement can be. This work present improved method of results' evaluation, so mistakes are better identified.

Language mode adaptation is often discussed because of it's dependence on various factors and different results. Several experiments are performed to demonstrate influence of the adaptation on recognizer developed in SpeechLab.

Finally, automatic punctuation approach is presented. Punctuation increases readability of recognizer's output. Presented approach uses only output of the SpeechLab's recognizer, because it's output also includes information of various noises.

Obsah

Seznam tabulek

Seznam obrázků

Kapitola 1

Úvod

Prudký rozvoj hlasových technologií v posledních desetiletích je z velké části zapříčiněn výrazným nárůstem výkonu výpočetní techniky, neboť zpracování přirozené řeči, zejména její rozpoznávání, je výpočetně velice náročné. Nemalý vliv má též velké množství publikovaných prací a jistá stabilizace postupů zpracování řeči.

V současné době se lze již setkat s množstvím aplikací využívajících zpracování řeči. Syntéza řeči je používaná v dialogových a navigačních systémech. Důležitou aplikací syntézy řeči jsou systémy pomáhající nevidomým jako je například čtečka obrazovky.

Automatické rozpoznávání mluvené řeči se též aplikuje v dialogových systémech. Příkladem je systém Infocity zahrnující jak syntézu tak i rozpoznávání. Infocity je dialogový telefonní systém podávající informace o Liberci z oblasti dopravy, kultury, sportu, atd. Tento systém byl vyvinut v Laboratoři počítačového zpracování řeči Technické univerzity v Liberci. Další aplikací rozpoznávání mluvené řeči jsou programy umožňující hlasové ovládání počítače a jednoduché diktování. Nejznámější jsou Dragon NaturallySpeaking od firmy Nuance Communications, ViaVoice od IBM a SpeechMagic od firmy Phillips, která se specializuje na rozpoznávání řeči v lékařské oblasti. Pro rozpoznávání češtiny byl vyvinut systém MyVoice [1], který má pomoci zejména handicapovaným lidem v přístupu k výpočetní technice a informačním technologiím. MyVoice pochází z Laboratoře počítačového zpracování řeči Technické univerzity v Liberci a je prodáván firmou Fugasoft.

Rozsáhlejší systémy, které zahrnují rozpoznávač mluvené řeči jsou používány při přepisu televizních a rozhlasových pořadů [2]. Takový systém byl vyvinut v Laboratoři počítačového zpracování řeči. Díky segmentaci vstupního akustického signálu je rozpoznávání distribuováno na více počítačů, čímž je dosažena přijatelná odezva celého přepisovacího systému. Automaticky přepsané zprávy jsou manuálně opravovány. Tento systém byl vyvinut pro firmu Newton IT.

Pravděpodobně nejrozšířenější aplikací rozpoznávání řeči je hlasové vytáčení v mobilních telefonech. Mnoho lidí však hlasové vytáčení nepoužívá z důvodů vysoké nepřesnosti rozpoznávání a citlivosti na změny prostředí.

Na závěr výčtu uplatnění hlasových technologií je nutné podotknout, že dosavadní aplikace jsou zatím orientovány především na spotřební trh s cílem ulehčit práci spotřebiteli, kde případná chyba aplikace nemá vážné důsledky.

Přestože je již oblast automatického rozpoznávání mluvené řeči zkoumána dlouhou dobu, není zatím možné používat hlasové technologie tak pohodlně, jak bychom si přáli. Překážky pro masové rozšíření hlasových technologií jsou:

- Vysoká citlivost na prostředí, ve kterém je řeč rozpoznávána. Přestože použití kepstrálních příznaků a Skrytých Markovských modelů snižuje citlivost rozpoznávače na prostředí, může být úspěšnost rozpoznávače v zarušeném prostředí výrazně nižší než v nahrávacím studiu nebo běžné kanceláři. Nestabilita úspěšnosti rozpoznávače řeči přispívá k nerozhodnosti uživatelů zaplatit za diktovací systém např. Dragon NaturallySpeaking přibližně 200 dolarů.

- Zaškolení uživatelů, aby mluvili plynule a nesnažili se různě intonovat, křičet či hláskovat, když rozpoznávač dělá chybu. Pro diktování izolovaných slov je potřeba dávat pozor na oddělování slov pauzami zvláště u předložek, které se běžně vyslovují společně s následujícím slovem.

- Způsob rozpoznávání řeči pomocí Viterbiho dekodéru je pro většinu jazyků podobný (některé implementace jsou volně k dispozici: Julius [3], HTK [4]). Problémem je však lokalizace rozpoznávače, zejména akustického a jazykového modelu a slovníku, neboť je manuálně, časově i finančně náročná a je nutno ji provést pro každý jazyk zvlášť. Hlasové technologie jsou proto nejvíce využívány pro nejvýznamnější světové jazyky jako je angličtina, japonština, španělština, atd. Využití rozpoznávání řeči pro ostatní jazyky je vázáno na aplikace výzkumu lokálních univerzit.

Rozpoznávání mluvené řeči se skládá z moha úkonů, které lze rozdělit do tří základních vrstev.

Akustická vrstva se stará o nahrání a zpracování řeči do podoby příznaků vhodných pro rozpoznávání. Cílem této vrstvy je potlačit nežádoucí složky akustického signálu, jako je šum a různorodost řečníků.

Technická vrstva zahrnuje rozpoznávací proces, kdy se k signálu přiřazuje nejpravděpodobnější sekvence hlásek, které tvoří slova.

Lingvistická vrstva vystihuje zákonitosti jazyka, který je rozpoznáván, a tím pomáhá technické vrstvě v efektivnějším prohledáváním variant přiřazení sekvence hlásek a nalezením nejlepší promluvy. Lingvistická vrstva též vkládá interpunkci do rozpoznané promluvy, aby se zvýšila čitelnost výstupu rozpoznávače.

Lingvistická vrstva je tedy nejproblematičtější částí lokalizaci rozpoznávače řeči. Jelikož mohou být jazyky, kterými lidé mluví, velmi rozdílné, jsou rozdílné i metody lokalizace. Některé jazyky mají výrazně méně slov než jiné, proto pro ně postačují menší slovníky a rozpoznávání není tudíž tolik výpočetně náročné. Různými slovy se v oblasti rozpoznávání řeči myslí i různé varianty jednoho slova vzniklé skloňováním, časováním, či jiným rodem slova. Lingvistická vrstva není navíc časově stabilní, neboť se objevují stále nová slova nebo kontexty slov již známých. Lingvistická vrstva je variabilní i v závislosti na aplikačním zaměření, jako je přepis nadiktovaných lékařských zpráv a soudních rozsudků.

Lokalizace lingvistické vrstvy spočívá v provedení řady úloh. Nejprve je nutné získat dostatečné množství dat a vytvořit tak textový korpus, ze kterého lze odvozovat další kroky. Pro každý jazyk je nutné vytvořit slovník obsahující slova, která je schopen rozpoznávač rozpoznat. Pro každé slovo ze slovníku se určí jeho výslovnost, neboli fonetický přepis, který slovo napojuje na akustický signál. Z textového korpusu a slovníku se odvodí jazykový model popisující závislosti mezi slovy. Po rozpoznání promluvy je nutné do výstupu z rozpoznávače přidat interpunkci z důvodu vyšší čitelnosti.

Tato práce zahrnuje komplexně pojatý problém lokalizace lingvistické vrstvy pro mluvenou češtinu. Postupy uvedené v této práci souvisí s vývojem skutečného systému rozpoznávání mluvené češtiny. V průběhu vytváření práce byl tento systém nasazen při přepisu zpravodajských pořadů. Jako modelový příklad jiné aplikace je uvedena úloha diktování lékařských zpráv.

Kapitola 2

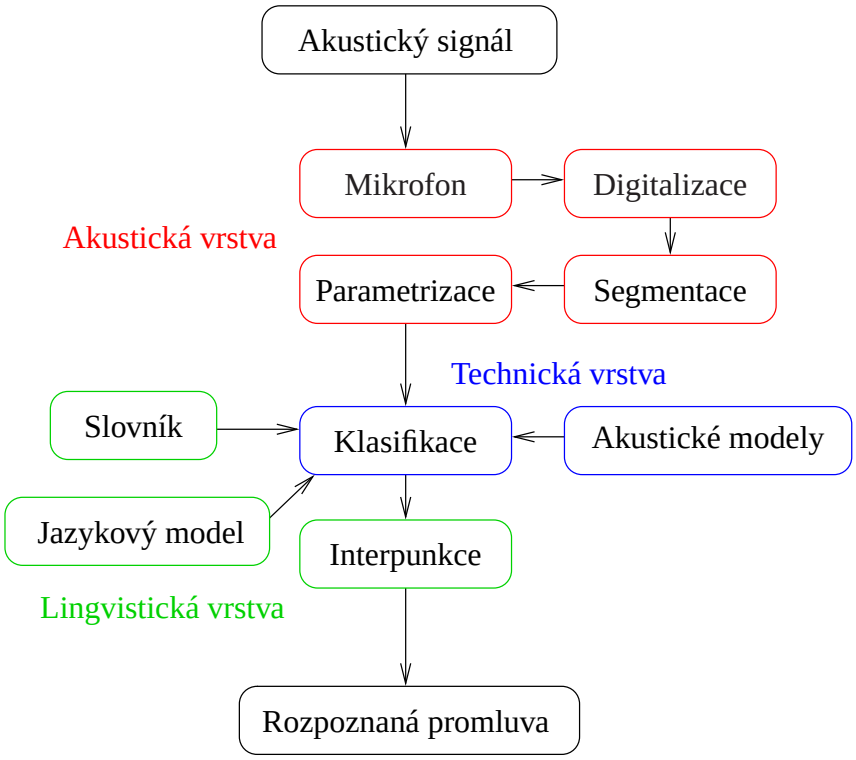
Principy rozpoznávání řeči a současný stav

2.1 Principy automatického rozpoznávání řeči

Před rozpoznáváním řeči je nutné nahraný signál obsahující řeč vhodně předzpracovat, aby obsahoval pouze informace podstatné pro rozpoznávání řeči. Toto předzpracování je v této práci nazýváno akustickou vrstvou. V rámci akustické vrstvy je odstraňován šum a potlačována různorodost mluvčích tak, aby bylo rozpoznávání na mluvčím minimálně závislé. Nahraný akustický signál je digitalizován ve zvukové kartě. Digitalizace spočívá ve vzorkování signálu a následné kvantizaci vzorků pomocí analogově-číslicového převodníku. Vzorky jsou dále rozděleny na krátké segmenty o délce 25 ms, které se nazývají framy. Sousední framy se vzájemně překrývají. Délka framu se volí tak, aby bylo možné považovat signál v rámci framu za stacionární. Dalším krokem zpracování signálu je parametrizace, která převede framy na příznaky splňující následující požadavky. Pomocí příznaků by mělo být možné jednoduše identifikovat jednotlivá slova ve slovníku. Zároveň by měly potlačit vliv různých řečníků (výška hlasu, síla signálu). Dále by příznaků nemělo být mnoho a měly by být jednoduše vypočítatelné. Nejpoužívanějšími příznaky jsou MFCC příznaky [5] pro nízkou citlivost na šum v signálu. Tyto příznaky jsou taktéž používány v rozpoznávačích použitých v této práci. Signál je pro další zpracování reprezentován maticí velikosti $N \times M$, kde N je počet framů a M je počet příznaků reprezentujících každý frame. Existuje množství přístupů pro zpracování signálu pro rozpoznávání řeči. Přehled základních metod lze nalézt v [5].

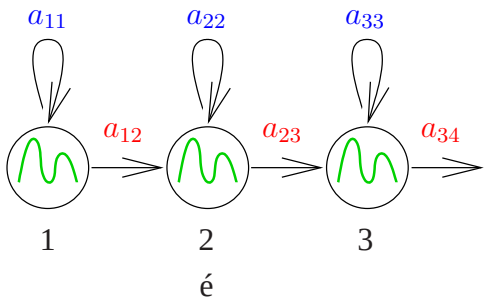
Rozpoznávání řeči se dělí na dvě základní úlohy a to rozpoznávání izolovaných slov a rozpoznávání spojité řeči. Rozpoznávání izolovaných slov je lehčí varianta. Cílem rozpoznávání izolovaných slov je přiřadit nahranému zvuku právě

jedno nejpravděpodobnější slovo. Součástí každého rozpoznávače izolovaných slov musí být detektor začátku a konce slova. Tento detektor využívá znalosti energie signálu reprezentované jedním z příznaků vytvořených v akustické vrstvě rozpoznávání. Rozpoznávání spojité řeči se k dané nahrávce snaží najít nejpravděpodobnější sekvenci slov. Není předem známo, kolik slov nahrávka obsahuje, v jakém pořadí jsou slova vyslovena a ani hranice slov není známa. Tato úloha má exponenciální složitost. Základní schéma rozpoznávání řeči je na obrázku 2.1.



Obrázek 2.1: Etapy rozpoznávání mluvené řeči

Výrazný průlom v rozpoznávání řeči zapříčinilo používání skrytých markovských modelů (HMM) pro reprezentaci slov a hlásek. Skryté markovské modely jsou dodnes využívány v naprosté většině systémů rozpoznávání mluvené řeči [6]. Podobně jako v analýze přirozené řeči je dobré identifikovat základní jednotky, ze kterých se řeč skládá. Vhodnými jednotkami jsou fonémy, neboť se jedná o základní stavební prvky řeči a je jich relativně málo, což je výhodné z hlediska výpočetní náročnosti. V rozpoznávání řeči je každý foném reprezentován skrytým markovským modelem, nejčastěji třístavovým. Příklad reprezentace fonému je uveden na obrázku 2.2. Všechny fonémy jsou reprezentovány třístavovým levo-pravým skrytým markovským modelem, liší se však v hodnotách parametrů přechodů a_{ij} a parametrů výstupní funkce. Parametry a_{ii} vyjadřují pravděpodobnost



Obrázek 2.2: Reprezentace fonému 3stavovým skrytým markovským modelem

setrvání ve stavu i, parametry a_{ij}, kde $i \neq j$, vyjadřují pravděpodobnost přechodu ze stavu i do stavu j. Výstupní funkce každého stavu ohodnocuje framy signálu přiřazené danému stavu. Framy jsou vytvořeny z nahrávky akustickou vrstvou. Jako výstupní funkce se většinou používá hustota vícemixturového normálního rozdělení daná vztahem

$$\sum_{m=1}^{M} c_m \frac{1}{\sqrt{(2\pi)^P \det \mathbf{\Sigma}_m}} \exp\left(-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}}_m)^T \mathbf{\Sigma}_m^{-1}(\mathbf{x}-\bar{\mathbf{x}}_m)\right), \qquad (2.1)$$

kde $m \in M$ je počet mixtur, P je počet příznaků reprezentujících jeden frame, $\mathbf{\Sigma}_m$ je kovarianční matice vektorů příznaků přiřazených danému stavu, $\bar{\mathbf{x}}$ je střední hodnota vektorů příznaků v daném stavu, $c_m \in < 0, 1 >$ je váha mixtury. Mixtury vícemixturového normálního rozdělení zachycují variabilitu hlásek vyslovovaných v různých kontextech různými lidmi v různém prostření.

Variabilitu fonémů lze též vyjádřit tak, že se jako základní jednotka nezvolí pouze jeden foném, ale i jeho okolí, například předchozí a následující foném. Vznikne takzvaný trifon. Počet základních jednotek se ale výrazně zvýší a je nutné získat více dat na spolehlivý odhad parametrů modelu.

Větší jazykové celky, jako jsou slova a věty, jsou vytvořeny zřetězením modelů fonémů. Proto je třeba znát fonetický přepis slov.

Z předchozího popisu modelování fonémů je patrný veliký počet parametrů. Každý foném je reprezentován 3stavovým modelem, což znamená 3 pravděpodobnosti přechodů mezi stavy, 3 pravděpodobnosti setrvání ve stavu. Každá mixtura má P středních hodnot a P diagonálních hodnot kovarianční matice. V reálných případech se nepočítá s plnou kovarianční maticí, nýbrž jen se zjednodušenou diagonální maticí z důvodu výrazně nižší výpočetní náročnosti a větší numerické stability inverze kovarianční matice. Pro češtinu je v Laboratoři počítačového zpracování řeči používáno až 100 mixtur pro jeden foném.

Odhad parametrů neboli trénování modelů se provádí z nahrávek, u kterých je známa sekvence fonémů v nich obsažených. Čím méně je k dispozici tréno-

vacích dat, tím preciznějí musí být proveden přepis nahrávek na fonémy. Preciznější přepis též zajišťuje rychlejší odhad parametrů. Odhad parametrů skrytých markovských modelů lze provést mnoha způsoby, např. Baum-Welchovým algoritmem [4, 5], simulovaným žíháním [7] nebo samoorganizujícími mapami [8]. Jiná, jednodušší metoda je uvedena dále. Pro jednoduchost je uvažována výstupní funkce jako hustota jednorozměrného unimodálního normálního rozložení. Trénování parametrů modelů je iterativní algoritmus, ve kterém se neustále zlepšuje přiřazení framů signálu jednotlivým stavům.

1. Pro všechny trénovací nahrávky se provede jejich fonetický přepis, čímž se zjistí sekvence fonémů v těchto nahrávkách.

2. Vytvoří se modely promluv zřetězením modelů příslušných fonémů.

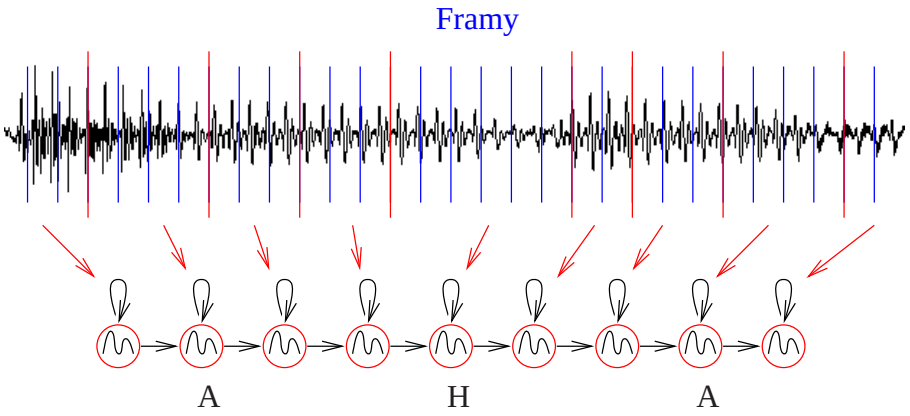
3. Framy signálu se rozloží postupně rovnoměrně mezi jednotlivé stavy fonémů.

4. Pro každý stav, tedy framy přiřazené ke stavu, se určí střední hodnota $\mu = \frac{1}{N_s}\sum_{i=1}^{N_s} x_i$ a rozptyl $\sigma^2 = \frac{1}{N_s}\sum_{i=1}^{N_s}(x_i - \mu)$. Pravděpodobnosti přechodů se určí jako $a_{ss+1} = \frac{K}{N_s}$ a $a_{ss} = 1 - a_{ss+1}$. Kde N_s je počet framů přiřazených danému stavu ze všech trénovacích nahrávek a K počet možných přechodů do následujícího stavu ve všech trénovacích nahrávkách. Pro levo-pravé modely uvedené na obrázku 2.2 je pouze jeden možný přechod do následujícího stavu. Proto K je též rovno počtu fonémů v trénovacích datech náležících HMM, pro který se určují parametry.

5. Pomocí dynamického programování se přeuspořádají framy tak, aby trénovací promluvy byly generovány s maximální pravděpodobností. Framy musí neustále následovat za sebou tak, jak byly vytvořeny v akustické vrstvě. Pravděpodobnost generovaných promluv je dána upravenými parametry modelů a_{ij}, μ a σ. Přiřazení framů stavům skrytých markovských modelů je naznačeno na obrázku 2.3.

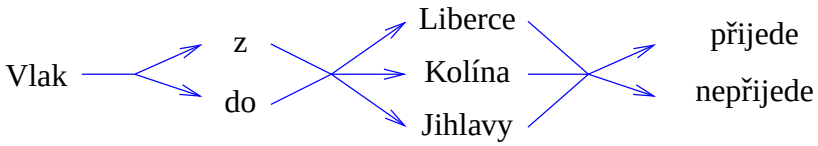
6. Opakujeme od kroku 4, dokud se mění parametry skrytých markovských modelů.

Převod textu na fonémy je jazykově závislý. Pro některé jazyky, například pro angličtinu, je převod obtížný, neboť neexistují jednoduchá spolehlivá pravidla, která by tuto úlohu automatizovala. Fonémy v češtině odpovídají přibližně písmenům. Převod napsaného slova na sekvenci fonémů (fonetická transkripce) lze pro česká slova vyjádřit ve formě pravidel, která jsou všeobecně známá a lze je nalézt v [9]. Nová a méně častá pravidla fonetické transkripce je možné doučit [10].



Obrázek 2.3: Přiřazení framů stavům skrytých markovských modelů

S přibývajícím počtem slov ve slovníku si začínají být jednotlivá slova navzájem akusticky blízká, proto je výhodné využít informaci o výskytu slov a slovních spojení v daném jazyce. Tuto informaci do rozpoznávače dodává jazykový model. V současné době jsou nejpoužívanější dvě formy jazykového modelu. První je založen na pevné gramatice. Je předem dáno, které sekvence slov se mohou vyskytovat a které nikoli. Příklad je uveden na obrázku 2.4. Výhodou pevné gra-



Obrázek 2.4: Pevná gramatická síť

matické sítě je jednoduchost, menší výpočetní náročnost a vyšší úspěšnost rozpoznávání v systémech s malým slovníkem. Vhodné nasazení tohoto jazykového modelu je v dialogových informačních systémech. Ve složitějších systémech se stává pevná gramatická síť složitou. Je-li sítí povoleno libovolné pořadí slov, pak nemají jednotlivé promluvy různé ohodnocení a jazykový model ztrácí pro rozpoznávání smysl. Pevnou gramatickou síť lze použít na zarovnání známé promluvy se signálem a určit tak časy hranic slov a fonémů.

Druhý typ jazykového modelu je pravděpodobnostní, kdy je povolena libovolná sekvence slov ze slovníku, ale tato sekvence je ohodnocena v závislosti na pravděpodobnosti výskytu v daném jazyce. Tento model se nazývá n-gramový. Pro každou sekvenci slov $w_n w_{n-1} w_{n-2} \dots w_1$ je určena podmíněná pravděpodobnost $P(w_n|w_{n-1}w_{n-2}\dots w_1)$, že následuje slovo w_n, pokud se již vyskytla sekvence $w_{n-1}w_{n-2}\dots w_1$. Tato pravděpodobnost se nazývá n-gram a je odhadována z velkého textového korpusu. Délka historie n je fixní, většinou 2 (bigramový

model) nebo 3 (trigramový model). Pravděpodobnost sekvence slov $w_1 w_2 \dots w_i$ je pak

$$P(w_1 w_2 \dots w_i) = P(w_i | w_{i-1} w_{i-2} w_{i-n+1}) P(w_{i-1} | w_{i-2} w_{i-3} w_{i-n+1}) \dots \quad (2.2)$$

N-gramový jazykový model je vhodný pro běžné univerzální systémy rozpoznávání mluvené řeči. V aplikacích jsou většinou n-gramy odhadovány maximálně věrohodným odhadem pomocí relativních četností výskytu slov z velkého textového korpusu. V případě, že se nějaká sekvence slov v korpusu nevyskytuje, byla by pravděpodobnost promluvy obsahující tuto sekvenci nulová. Tomu se předchází takzvaným vyhlazování, kdy jsou nulové n-gramy nahrazeny malým číslem. Vyhlazovacích technik existuje několik [5]. Nejednodušší je přičíst ke všem výskytům sekvencí slov 1. Výhodou n-gramů oproti pevné síti je univerzálnost použití. N-gramový model je též vhodný pro velké slovníky.

Jakmile je již k dispozici akustický model ve formě HMM a jazykový model ve formě n-gramů, je možné hledat nejlepší sekvenci slov odpovídající nahrané promluvě. Rozpoznávání (dekódování) sekvence lze provést pomocí časově synchronního Viterbiho dekodéru, který je založen na dynamickém programování. Viterbiho dekodér hledá takovou cestu skrze stavy modelů, která má největší pravděpodobnost. Pravděpodobnost cesty skrze stavy je dána pravděpodobnostmi setrvání ve stavu a_{ss}, pravděpodobnostmi přechodů do následujícího stavu a_{ss+1}, hodnotami výstupních funkcí pro framy ve stavech (pravděpodobností generování těchto framů) a n-gramy (pravděpodobnosti přechodu mezi slovy) v případě přechodů mezi slovy. S každým novým framem se prodlouží cesta o 1 krok. Musí být tedy provedeno setrvání ve stavu, nebo přechod do jiného stavu. Cena cesty je součinem pravděpodobností, které byly získány při provedení jednotlivých kroků. Pravděpodobnosti přechodů lze též chápat jako cenu přechodu. Hledá se pak cesta s maximální cenou. Časově synchronní Viterbiho algoritmus též musí brát v úvahu absenci informace o hranici slov, proto je nutno předpokládat, že s každým novým framem může začít nové slovo, nebo nějaké skončit. Jednoduše popsat tento komplexní algoritmus ve velice složité. Algoritmus je popsán v [6]. Schéma nejpravděpodobnější sekvence slov je uvedeno na obrázku 2.5.

Viterbiho algoritmus je založen na dynamickém programování, které hledá globální optimum (cestu s maximální cenou, nejpravděpodobnější sekvenci slov). Přestože se jedná o velmi efektivní algoritmus hledání globálního optima, je nutné zjistit ceny všech možných cest, jejichž počet exponenciálně stoupá s velikostí slovníku a délkou rozpoznávané nahrávky. Nejpravděpodobnější nejlepší cesta se dá zjistit zpětným procházením až po zjištění cen všech cest. Je tedy zřejmé, že pro přijatelnou odezvu v čase je nutné použít prořezávání a procházet pouze nejslibnější cesty. Algoritmus s prořezáváním již nemusí najít optimální cestu. V praxi se však ukazuje, že lze bez extrémně náročného testování nalézt kompromis mezi

Obrázek 2.5: Nepravděpodobnější sekvence slov

množstvím nenavštívených málo perspektivních cest a snížením úspěšnosti rozpoznávání.

Z výše uvedených principů rozpoznávání přirozené řeči je patrné, že principy rozpoznávání spoléhají na množství parametrů jednotlivých modelů, které je třeba co nejpřesněji odhadnout. Pro zjištění parametrů akustických modelů ve formě HMM je nutné provést pečlivý ruční přepis několika hodin různorodých nahrávek. Spolehlivý odhad n-gramů je možný z velkého textového korpusu obsahujícího gigabyty dat. Zatímco počet fonémů je v dané řeči v podstatě neměnný a relativně malý, množství různých slov se svými variantami je značné, a proto vytvoření kvalitního textového korpusu může trvat dlouhou dobu.

2.2 Metody vývoje lingvistické vrstvy

2.2.1 Tvorba textového korpusu

Tvorba lingvistické vrstvy vyžaduje veliké množství textu na jeho důkladnou analýzu. Textový korpus musí dostatečně reprezentativně pokrývat požadovanou aplikační oblast, aby z něj vytvořený slovník rozpoznávače pokryl co nejvíce nejčetnějších slov. Problémem je, že zdroje textu pro různé aplikace se mohou výrazně lišit co do přístupnosti, požadovaného rozsahu, či míry znečištění například překlepy a zkratkami.

Nejčastějším a nejpřístupnějším zdrojem textu jsou webové stránky. Pro automatizaci stahování webových stánek existuje množství nástrojů, od specializovaných programů na vytváření zrcadel portálů, jako například program *wget*, po specializované knihovny programovacích jazyků, například knihovna *LWP* pro jazyk *Perl*, která je dobře popsána v [11]. *LWP* umožňuje transformaci webo-

vých stránek do stromové struktury, a tím i snadné vyhledávání relevantních odkazů na další stránky. Stromovou strukturu lze také úspěšně použít k extrakci těch částí stránky, které obsahují užitečný text. Podobné knihovny existují i pro jiné programovací jazyky jako Python, Java, atd. Sběr dat z webu provádějí také webové vyhledávací služby. Například Google prohledává internet pomocí programu Geocrawler. Sběr dat z webu pro textový korpus může probíhat necíleně, náhodným procházením odkazů, nebo cíleně například pro aplikaci rozpoznávání zpráv z televize a rádia [12] lze sbírat novinové články z webových portálů denního tisku. Necílený přístup je schopen v krátkém čase získat velké množství textu obsahujícího ale množství nežádoucích slov, jako jsou například slova jiného než požadovaného jazyka. Pomocí cíleného prohledávání požadovaných webových portálů lze získat menší množství textu, neboť je ho denně napsáno jen nepatrné množství v porovnání s požadavky na kvalitní korpus. Proto musí stahování probíhat delší čas.

Dalším zdrojem dat pro přepis zpráv mohou být přepisy pořadů od firem zabývajících se touto činností. Těchto dat není velké množství, ale jedná se o přepisy mluvené řeči, tedy přímo o data nejlépe pokrývající cílovou oblast. Přepisy zpráv mají významný vliv na zvyšování úspěšnosti přepisu zpravodajských pořadů, neboť obsahují používané promluvy. Přepisy pořadů jsou čisté a je tudíž jednodušší z nich přidat chybějící slova do slovníku, a tím umožnit jejich rozpoznání. V české republice se přepisem a monitoringem zpráv zabývají firmy Newton IT a Anopress IT.

Problematické je sehnat data pro speciální aplikace, jako jsou přepisy lékařských zpráv nebo rozsudků. Tato data obsahují osobní informace, proto nemohou být bez jejich odstranění zpracovávána. Odstranění osobních informací není jednoduchá záležitost, neboť jsou často zakomponována přímo do textu. Pak nejsou tyto zdroje přístupné vůbec. Dalším problémem zdrojů specializovaných dat je veliké množství překlepů a oborových zkratek, což vzhledem k malému množství dat vyžaduje pečlivou a nákladnou úpravu textů pro další zpracování. Metody čištění textů jsou přímo vázané na zdroje dat.

2.2.2 Tvorba slovníku

Pokud je již k dispozici dostatečně velký textový korpus, lze s jeho pomocí vytvořit slovník pro rozpoznávač řeči. Tento slovník obsahuje slova a jejich fonetické přepisy. Jako samostatná slova jsou v rozpoznávání řeči chápány i varianty vytvořené od jednoho kořene slova například skloňováním, časováním, různým rodem atd. Například v češtině *řidič, řidiče, řidiči, řidičovi, řidička, řidičce* jsou různá slova. Jazyky s tímto způsobem vytváření slov se nazývají inflektivní. Míra inflektivity je v každém jazyce různá. Minimální je například v angličtině, značná v češtině a němčině. Velmi inflektivní jazyky vyžadují výrazně větší slovníky, což

má za následek vyšší výpočetní nároky a větší množství dat ke spolehlivému odhadu parametrů. Existují i náročnější jazyky na velikost slovníku, takzvané aglutinativní jazyky, kde jedno slovo ve slovníku pro rozpoznávač je složenina více slov, či dokonce část věty nebo fráze. Mezi tyto jazyky patří němčina, finština a turečtina. V němčině nebo holandštině se tento problém řeší dekompozicí složenin [13, 14], ve finštině je snaha vytvořit slovník z menších jednotek, například morfémů [15]. Problém se slovníkem založeným na subslovních jednotkách je, že není známo, kde je začátek a konec slova. Dalším problém subslovních jednotek je jejich identifikace. Ruční dekompozice je náročná a není vždy jednoznačná. Jednotky dekomponované automaticky [16] nemusí odpovídat jednotkám používaným v lingvistice a jejich interpretace je pak nejasná.

Slovník je ve většině případů tvořen nejčetnějšími slovy z textového korpusu. Odlišný přístup používaný pro inflektivní jazyky spočíval v přidání nejčetnějších slov a všech jejich odvozenin. Tento přístup byl používán, pokud nebyl k dispozici dostatečně veliký textový korpus. Nevýhoda generování tvarů slov spočívala v občasné nepravidelnosti při generování tvarů a značném zvětšování slovníku.

Slova, která nejsou ve slovníku, se často označují jako OOV. Pokud není slovo ve slovníku, není možné, aby jej rozpoznávač rozpoznal. Vzhledem k vlastnostem Viterbiho algoritmu je místo slova chybějícího ve slovníku rozpoznáno podobné slovo nebo sekvence slov ze slovníku. Je tedy zřejmé, že pokud se v nahrávce vyskytuje slovo, které není ve slovníku, dojde k alespoň jedné chybě, často i k sekvenci chyb rozpoznávání.

Pokud rozpoznávač nepodporuje veliké slovníky, je možné provést adaptaci při druhém průchodu rozpoznávání, kdy se do slovníku přidají slova se stejným kořenem slova, jako mají slova rozpoznaná v prvním průchodu. Tak lze eliminovat chyby typické pro inflektivní jazyky, kdy se tvary slov liší jen málo [17].

Na základě analýzy výsledků rozpoznávání bylo zjištěno, že krátká slova jako předložky a spojky jsou často vypouštěna nebo přidávána. Proto je vhodné krátká slova spojit se sousedícím větším slovem a vytvořit tak jedno delší slovo. Tento přístup vede ke snížení chybovosti způsobené právě krátkými slovy [18, 19]. Je však nutné opatrně zvolit způsob spojování slov, aby nedošlo k nepřiměřenému zvětšení slovníku.

2.2.3 Fonetický přepis slov

Fonetická transkripce slov spojuje akustickou a textovou část rozpoznávání. Úkolem fonetické transkripce je namapovat písmena na fonémy, tedy slova na jejich skryté markovské modely. Pro zápis fonémů je možno použít mezinárodní fonetickou abecedu (IPA) popsanou v [20]. Pro češtinu byla vypracována abeceda PAC [21], která přehledněji zachycuje čeká pravidla pro fonetický přepis, neboť fonémy přibližně odpovídají písmenům. Česká fonetická abeceda je i implemen-

tačně výhodnější, neboť každý český foném je reprezentován jedním znakem. Pro každé slovo ve slovníku je třeba vytvořit jeho fonetický přepis včetně výslovnostních variant. Například slovo *osm* lze vyslovovat jako *osm* nebo *osum*. Jelikož slovníky obsahují desetitisíce i sta tisíce slov, je ruční fonetický přepis obtížný. Složitost automatické fonetické transkripce závisí na jazyce, pro který je prováděna. Automatický přepis anglických slov je obtížnější než přepis českých slov, neboť v angličtině neexistují jednoduchá fonologická pravidla, která by bylo možné implementovat. Fonologická pravidla pro automatický fonetický přepis mohou být implementována ve formě produkčních pravidel [22], [23], rozhodovacích stromů [24], konečných automatů [25] nebo neuronové sítě [26]. Pravidla je možné převzít z fonetiky daného jazyka, nebo odvodit z ručně přepsaných příkladů. Fonologická pravidla pro češtinu jsou uvedena v [9].

2.2.4 Jazykový model

V současných systémech rozpoznávání mluvené řeči dominuje n-gramový jazykový model, což je zejména dáno univerzálností jeho použití a dobrými výsledky. Parametry n-gramového modelu se odhadují z velkého textového korpusu. Pro každou sekvenci slov $w_n w_{n-1} w_{n-2} \dots w_1$ je určena podmíněná pravděpodobnost $P(w_n|w_{n-1}w_{n-2}\dots w_1)$. To znamená, že následuje slovo w_n, pokud se již vyskytla sekvence $w_{n-1}w_{n-2}\dots w_1$. Nejčastěji je používán bigramový a trigramový jazykový model, kde je zjišťována pravděpodobnost výskytu slov v závislosti na jednom, respektive dvou předchozích slovech.

Kromě slov může být n-gramový jazykový model založen i na morfologických třídách [27]. Místo sekvencí slov je pak odhadována podmíněná pravděpodobnost třídy v závislosti na sekvenci předchozích tříd. Textový korpus musí být též převeden na třídy, což není pro jazyky s volným pořádkem slov ve větě (například čeština) triviální operace. Naopak v angličtině a němčině s pevným pořádkem slov je situace jednodušší, neboť je známé, kde se nachází podmět, kde přísudek, atd. Automatický převod korpusu na třídy se nazývá tagování. Pro češtinu je způsob tagování uveden v [28]. Slovníky jsou i pro třídní n-gramové modely složeny ze slov. Proto je třeba převést třídní jazykový model na slovní jazykový model. To lze pomocí vztahu

$$P(w_n|w_{n-1}w_{n-2}\dots w_1) = P(w_n|c_n)P(c_n|c_{n-1}c_{n-2}\dots c_1), \qquad (2.3)$$

kde w_i jsou slova a c_i třídy. Ze vztahu 2.3 je patrné, že oproti slovnímu n-gramovému modelu je třeba odhadnout další parametry $P(w_n|c_n)$. Tříd je však mnohem méně než slov, proto může třídní n-gramový model ve výsledku obsahovat méně parametrů než slovní.

Problémem n-gramového modelu jsou odhadnuté nulové pravděpodobnosti, pokud se sekvence slov v korpusu nevyskytuje. Pokud promluva obsahuje nevy-

skytující se sekvenci, pak je pravděpodobnost celé promluvy nulová, což plyne ze vztahu 2.2. Nahrazení nulových pravděpodobností nenulovými se nazývá vyhlazování. Existuje několik metod vyhlazování uvedených například v [5]. Nejjednodušší metoda vyhlazování je přečtení 1 ke každé absolutní četnosti výskytu sekvence slov. Bigram je pak dán následujícím vztahem

$$P(w_n|w_{n-1}) = \frac{c(w_{n-1}w_n)+1}{c(w_{n-1})+V}, \tag{2.4}$$

kde V je počet slov ve slovníku a $c()$ jsou absolutní četnosti výskytu. Velmi používaná metoda je Witten-Bell [29] daná vztahem

$$P(w_n|w_{n-1}) = \frac{c(w_{n-1}w_n)}{c(w_{n-1})+N(w_{n-1})}\text{ , když } c(w_{n-1}w_n)>0 \tag{2.5}$$

$$= \frac{N(w_{n-1})}{(V-N(w_{n-1}))(c(w_{n-1})+N(w_{n-1}))}\text{ jinak , } \tag{2.6}$$

kde $N(w_{n-1})$ je počet různých následníků slova w_{n-1}.

Jazykový model pro veliké slovníky může být obtížné spočítat z důvodů velkých paměťových nároků. Bigramový jazykový model pro slovník obsahující 300000 slov může obsahovat až 300000^2 slovních dvojic, což v současné době nelze uchovat v paměti počítače. Ve skutečnosti je slovních dvojic viděno mnohem méně a lze je proto uchovat v paměti. Pro výpočet jazykového modelu existuje několik nástrojů, SRILM toolkit [30], CMU SLM [31]. Programy implementované ve skriptovacích jazycích nejsou vhodné pro výpočet jazykového modelu s velkým slovníkem, neboť vyžadují veliké množství paměti. Nejznámější software je SRILM, který je implementován v jazyce C. SRILM je univerzální soubor programů schopný spočítat různé typy jazykových modelů. Pro n-gramové modely je řád modelu omezen pouze velikostí dostupné paměti. Jsou též k dispozici různé metody vyhlazování jazykového modelu.

Správný jazykový model má reflektovat jazyk, kterým se mluví a který je následně rozpoznáván. Pokud chceme rozpoznávat tématické promluvy, jako je například jednání parlamentu, lékařské zprávy, sportovní přenosy, atd., je nutné vytvořit nový jazykový model nebo upravit existující. Většina technik adaptace jazykového modelu mixuje existující jazykové modely, přičemž minimalizují perplexitu nového jazykového modelu na testovacích promluvách. Přestože se daří výrazně snižovat perplexitu nových modelů, k znatelnému zlepšení úspěšnosti rozpoznávání s novými jazykovými modely dochází zřídka [32, 33, 34, 35]. V literatuře se objevuje více metod adaptace jazykového modelu, od nejjednodušší lineární interpolace [36], log-lineární interpolace [37], maximum a posteriori (MAP) adaptace vycházející z metod adaptace používané na akustické modely [38], adaptace založené na principu maxima entropie [39], po různé ad-hoc metody.

2.2.5 Úpravy výstupu rozpoznávače

Z rozpoznávače vychází proud mezerami oddělených slov, což je málo čitelné pro další zpracování. Vizuální přehlednost výsledků rozpoznávání významně podporuje interpunkce a velká písmena na začátku názvů tak, jak je to obvyklé v běžných textech. Velká písmena na začátku vět též přispívají ke zvýšení čitelnosti výsledků. Velká písmena na začátku slov jsou kromě počátků vět závislá na jazyce, tedy jazykovém modelu.

Automatická interpunkce se snaží najít konce vět a vložit do nich tečky a čárky v případě souvětí. K odhadnutí správné pozice interpunkce je třeba kombinovat informace z akustické části promluvy, jazykového modelu a morfologické analýzy. Detailní morfologická analýza je však závislá na znalosti pozic interpunkčních znamének. Morfologická analýza může být částečně nahrazena jazykovým modelem. Morfologická analýza významněji pomáhá v jazycích s pevným pořadím slov.

Z literatury je patrné, že dosavadní systémy provádějící automatickou interpunkci kombinují znalost průběhu základní frekvence (F0), n-gramového jazykového modelu, délky trvání fonémů [40] a případně i morfologických značek [41]. Průběh F0 je po částech linearizován a jsou z něj extrahovány různé příznaky, například sklon lineárních úseků. V práci [41] bylo pozorováno, že v češtině pozice čárek závisí spíše na informacích z jazykového modelu, zatímco pozice teček je spíše určena akustickou částí promluvy. Tentýž článek používá morfologický analyzátor k seskupení málo častých slov.

Kapitola 3

Cíle práce

3.1 Východiska

Tato práce je úzce spjata s vývojem systému pro automatický přepis televizních a rozhlasových pořadů a výsledky práce jsou v tomto systému uplatněny. Při vývoji rozsáhlého sytému pro rozpoznávání mluvené češtiny bylo třeba odpovědět na řadu koncepčních i dílčích otázek, vyřešit řadu dílčích úloh, implementovat je do modulů a tyto moduly správně propojit. Vzhledem k praktickému nasazení pak bylo též nutné řešit úlohy efektivní a paralelní správy slovníku a jazykového modelu a možnosti jejich časové adaptace.

Otázky, na které bylo třeba najít odpovědi:

Jak velký musí být slovník, aby dostatečně pokrýval češtinu? Je zřejmé, že pokud není slovo ve slovníku, není možné, aby jej rozpoznávač rozpoznal. Pokud se v promluvě vyskytne slovo, které není ve slovníku, udělá rozpoznávač chybu tím, že jej zamění za jiné podobné slovo ve slovníku. Často však rozpoznávač zamění chybějící slovo a jeho okolí sekvencí slov ze slovníku, což způsobí více chyb. Je zřejmé, že pro inflektivní jazyky s velikým počtem slov nebude možné pracovat s kompletním slovníkem všech slov, což je dáno zejména vysokými výpočetními nároky při používání velmi velkého slovníku. Podobná slova ve slovníku mají i podobné akustické modely, což vede k častým chybám v rozpoznávání. Veliké slovníky je též obtížné spravovat, a proto obsahují množství chyb jako jsou překlepy nebo špatné fonetické přepisy.

Z jakých zdrojů tento slovník tvořit? Velký slovník vyžaduje velké množství textu z dané aplikační oblasti. Z tohoto textu je odvozen jak slovník tak i jazykový model. Nejpřístupnějším zdrojem dat pro přepis zpráv jsou webové portály zpravodajských pořadů. Použití webu jako zdroje dat s sebou přináší

mnohé problémy. Je nutné vytvořit dostatečně robustní programy schopné pracovat 24 hodin denně, 365 dní v roce, neboť množství textu vytvořeného za jeden den není příliš veliké a navíc se často opakuje v různých zdrojích. Při stažení stránky je třeba zkontrolovat, jestli obsahuje požadované informace a provést extrakci podstatných dat, kterých může být na celé stránce i méně než třetina.

Jak předzpracovat výchozí text? Nasbíraný text obsahuje množství zkratek a číslovek, které je nutné rozepsat do tvaru více podobného jejich výslovnosti. Tento úkol není pro inflektivní jazyky jednoduchý, neboť přepis některých zkratek a číslovek je nutné vytvořit ve správném tvaru, což je někdy možné až po analýze okolí slova. Některé zkratky je naopak vhodné přidat do slovníku tak, jak jsou, a vytvořit pouze alternativní výslovnosti.

Je vhodné přidat do slovníku i slovní spojení? Slovní spojení ve slovníku je v mnoha inflektivních jazycích spíše problém, který pouze zvětšuje slovník a ředí data pro jazykový model. Na druhé straně je zřejmé, že krátká slova způsobují vyšší chybovost než slova dlouhá, proto je výhodné je spojit se sousedními slovy a vytvořit tak jedno slovní spojení zapsané ve slovníku jako jedno slovo. Tato úloha je spíše úlohou nalezení vhodného kritéria pro výběr slovních spojení.

Jak efektivně vytvořit výslovnost ke slovům? Slovník obsahuje veliké množství slov a jejich manuální fonetická transkripce je v přijatelné době nerealizovatelná. Proto je třeba použít a implementovat fonologická pravidla, která provedou automatickou fonetickou transkripci. V češtině se však vyskytuje množství slov cizího původu, na která nejsou česká fonologická pravidla aplikovatelná. Nejčastější problémy tvoří přepis slabik di, ti a ni. Pro tyto slabiky je třeba nalézt další pravidla tak, aby slova správně přepsaná českými pravidly nebyla poškozena a nová pravidla opravila co nejvíce chyb.

Jak upravit výstup rozpoznávače, aby byl co nejvíce čitelný? Výstup rozpoznávače je tvořen sekvencí mezerami oddělených slov, což je značně nečitelné. Automatická interpunkce a správná první velká písmena významně zvyšují čitelnost. Interpunkce je z části závislá na intonaci, tudíž na vstupním signálu. Při rozpoznávání je však automatická interpunkce posledním článkem, a tudíž od signálu velmi vzdálena.

Jak důležitá je pravidelná aktualizace slovníku a jazykového modelu? Je zřejmé, že se témata ve zprávách v čase mění. Je tudíž nutné provádět občasné aktualizace slovníku a jazykového modelu. Hlavní otázkou je jak často, neboť i tato operace zabírá čas, který může být při méně častých

úpravách využit efektivněji. Aktualizaci slovníku není totiž možné provádět zcela automaticky z důvodu velkého množství překlepů vybraných frekvenční analýzou za kandidáty na přidání.

Jak lze adaptovat lingvistickou vrstvu pro jinou aplikační oblast? Pokud je již k dispozici rozsáhlý systém pro přepis zpráv, je žádoucí, aby mohl být co nejsnadněji použit i v jiných aplikacích. Otázkou je, co bude nutné provést pro jeho adaptaci a kolik to bude stát.

3.2 Dílčí úlohy

Cílem této práce je tvorba lingvistické vrstvy pro rozpoznávač řeči s tím, že veškeré kroky jsou maximálně automatizovány. Výsledky výzkumu jsou aplikovány na rozpoznávač izolovaných slov [42] a spojité řeči [2] vyvíjené v Laboratoři počítačového zpracování řeči technické univerzity v Liberci. Oba rozpoznávače jsou primárně určeny pro rozpoznávání češtiny, čímž je také demonstrována lokalizace jazykového modelu a slovníku, a tedy snížení jedné z překážek masového rozšíření hlasových technologií.

Tvorba lingvistické vrstvy pro rozpoznávač mluvené češtiny zahrnuje mnoho akcí, z nichž některé jsou plně automatizovatelné, některé jen částečně a některé může udělat pouze manuálně specialista, například přepis lékařských zkratek. Při automatizaci jednotlivých akcí je nutné aplikovat jak hrubou výpočetní sílu, tak i heuristické informace a metody umělé inteligence. Hlavní úkoly při tvorbě lingvistické vrstvy jsou:

Tvorba textového korpusu: Pro rozpoznávání zpráv z televize a rádia [12] lze sbírat novinové články z webových stránek. Sběr dat z webu lze zajistit robustním programem automaticky prohledávajícím zvolené stránky. Při vytváření slovníku pro lékařský diktovací systém [43] je nutné z dat vypustit osobní informace pacientů, což komplikuje sběr dat.

Čištění a normalizace nasbíraných dat: Nasbíraná data obsahují číslicemi psané číslovky a zkratky, které je nutno rozepsat. U číslic se tak zmenší počet různých slov a zlepší se jazykový model. U zkratek se zjednoduší fonetický přepis. Expanze zkratek a číslovek není triviální, neboť je nutné vygenerovat správný tvar (pád), což nelze provést vždy automaticky. V některých případech pomůže automatická morfologická analýza [44]. Speciální zkratky mohou přepsat jen specialisté, kteří je používají. Čištění je operace výrazně závislá na jazyce a konkrétním zdroji textů.

Výběr slov do slovníku: Do slovníku se ze získaných textů vybírají nejčetnější slova jazyka. Se snižující se četností výskytu slov přibývá překlepů, cizích

a nesmyslných slov. Čeština jako inflektivní jazyk obsahuje mnoho tvarů slov, někdy i správné tvary mohou být méně četné než překlepy. Proto nelze proces výběru slov do slovníku plně automatizovat.

Vytvoření fonetické transkripce slov ve slovníku: Fonetická transkripce definuje napojení slova na akustické modely. Pro češtinu existuje soubor pravidel, která platí pro většinu českých slov. Cizí slova je nutné většinou přepisovat ručně. Jiné jazyky, například angličtina, mohou mít fonetickou transkripci obtížněji algoritmizovatelnou. Pro implementaci fonologických pravidel se používají produkční pravidla, stavové automaty nebo neuronové sítě.

Vytvoření jazykového modelu: Používaný rozpoznávač řeči používá jazykový model ve formě dvojic sousedních slov. Pro slovník obsahující 300000 slov je teoreticky možných 300000^2 slovních dvojic je proto nutné zabývat se implementací počítání těchto dvojic, aby je bylo možné umístit do paměti běžně dostupných počítačů.

Adaptace jazykového modelu: Při rozpoznávání televizních zpráv dochází v průběhu času ke změně témat. Proto je nutné aktualizovat slovník i jazykový model, což zahrnuje všechny předchozí akce, ale s menším množstvím dat a větším množstvím šumu v datech (překlepů). Adaptace jazykového modelu spočívá ve vhodném kombinování různých existujících jazykových modelů tak, aby výsledný model měl minimální perplexitu na testovacích datech.

Automatická interpunkce: Výstupem automatického rozpoznávače spojité řeči je mezerami oddělený proud slov. Pro zvýšení čitelnosti tohoto výstupu je nutné provést automatickou interpunkci zvýrazňující konce vět. Automatická interpunkce kromě akustické informace využívá též informaci z jazykového modelu.

Detailní analýza výsledků rozpoznávání: Pro efektivní zvyšování úspěšnosti rozpoznávání je výhodné vědět, která slova jsou nejčastěji špatně rozpoznávána. Běžně používaná metoda vyhodnocování výsledků rozpoznávání počítá slova, která jsou zaměněná, vložená, či vypuštěná. V případě výskytu sekvence chyb není běžnou metodou zjišťováno, které slovo je vložené a které zaměněné, je pouze zjištěno, že jedno je vložené a jedno zaměněné.

Kapitola 4

Systémy, nástroje a data využité při řešení

V této práci jsou prováděny experimenty na dvou typech rozpoznávačů vyvinutých v Laboratoři počítačového zpracování řeči na Technické univerzity v Liberci. První je rozpoznávač izolovaných slov schopný pracovat se slovníkem o velikosti až 1000000 slov [42]. Druhý rozpoznávač je navržen na rozpoznávání spojité řeči se slovníkem o velikosti až 400000 slov [2].

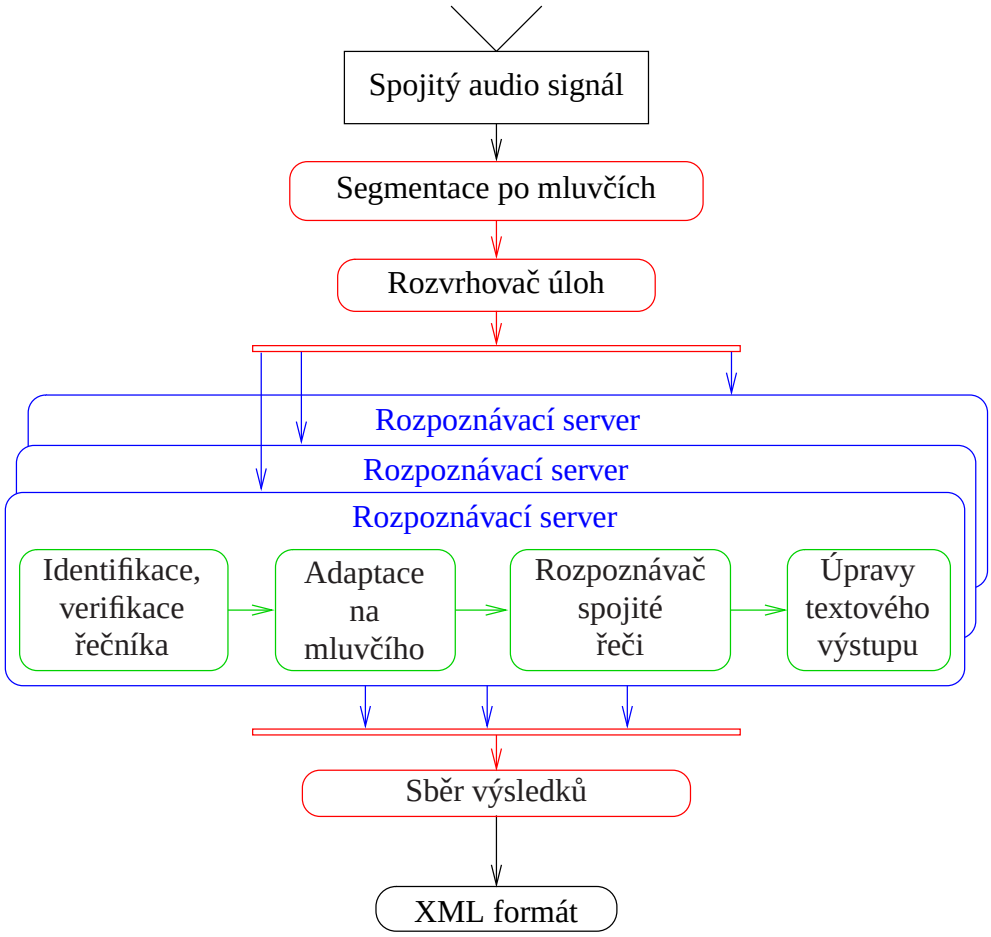
4.1 Systém automatické transkripce televizních a rozhlasových pořadů

V Laboratoři počítačového zpracování řeči Technické univerzity v Liberci byl vyvinut systém pro automatický přepis televizních a rozhlasových pořadů. Tento systém je velmi modulární, což umožňuje provádět množství různých experimentů. Systém je implementován tak, že rozpoznávače běží na několika počítačích najednou a úloha rozpoznávání je distribuována, čímž se zrychlí provádění experimentů na reálných datech. Zrychlením provádění experimentů také dochází k rychlejšímu vývoji v oblasti rozpoznávání řeči, neboť je možné provádět i experimenty, které nebyly realizovány díky obtížně predikovatelným výsledkům a velkým časovým nárokům.

Tento systém je nasazen v komerční sféře na přepis televizních a rozhlasových pořadů. Schéma systému je uvedeno na obrázku 4.1.

4.1.1 Zpracování signálu a extrakce příznaků

Vysílaný signál je zachycován běžnou televizní kartou a vzorkován 16 kHz v 16 bitovém rozlišení. Parametrizace je prováděna 100 krát za sekundu po 25 ms



Obrázek 4.1: Systém pro přepis televizních a rozhlasových pořadů

framech. Každý frame je reprezentován 40 příznaky: logaritmus energie signálu a 39 MFCC příznaky. Logaritmus energie je používán pouze k identifikaci řečové aktivity.

4.1.2 Segmentace signálu

Aby bylo možno úlohu distribuovat na více rozpoznávačů, je signál segmentován na akusticky homogenní úseky, kdy hovoří jeden mluvčí. Tímto způsobem jsou identifikovány i televizní znělky. Znělky nejsou dále zpracovávány rozpoznávačem řeči.

V poslední verzi je použita segmentace na základě vyhodnocení průběhu Bayesova informačního kritéria(BIC). Je využita metoda binárního dělení, kdy je průběh postupně hierarchicky dělen v bodě maxima od nejvyššího bodu až do

stanoveného prahu, hodnoty BIC [45].

4.1.3 Identifikace mluvčího

Ke každému segmentu je přiřazena informace o mluvčím. Pro účely rozpoznávání zpráv byl vytvořen seznam nejčastějších mluvčích a jejich modelů v podobě Gaussovských mixturových modelů (GMM). Nejdříve je identifikován mluvčí jako akusticky nejbližší model ze seznamu. Následně je identifikovaný mluvčí verifikován pomocí univerzálního modelu a potvrzen, či zamítnut. Pro zamítnuté mluvčí je segmentu přiřazena alespoň informace o pohlaví mluvčího, která je zjištěna na základě majority pohlaví nejbližších mluvčích ze seznamu.

4.1.4 Adaptace na mluvčího

Pro identifikované mluvčí byly připraveny adaptované akustické modely, které jsou při rozpoznávání řečového segmentu použity. Pro segmenty, kde je známa jen informace o pohlaví mluvčího, je provedena on-line adaptace kombinací akusticky nejbližších modelů mluvčích stejného pohlaví známých z identifikace mluvčího [46].

4.1.5 Rozpoznávač spojité řeči

Fonetická abeceda rozpoznávače obsahuje 41 českých fonémů a 7 symbolů pro šumy [47]. Každý symbol fonetické abecedy je modelován 3stavovým levopravým skrytým markovským modelem s vícemixturovou (až 100 mixtur na stav) výstupní funkcí. Akustické modely byly natrénovány na 35 hodinách anotovaných mikrofonních a vysílaných záznamů.

Rozpoznávání je založeno na jednoprůchodovém Viterbiho dekodéru. Rozpoznávač používá slovník obsahující 312 tisíc slov a 335 tisíc výslovnostních variant. Jazykový model je slovní bigramový. Primární vyhlazovací metoda je Witten-Bellova metoda. Pokud je vyhlazená hodnota neviděného bigramu vyšší než viděného je použita metoda Add-one.

4.1.6 Úpravy textového výstupu

Rozpoznávač produkuje řetězec slov oddělený mezerami. Pro zvýšení čitelnosti výstupu je provedena automatická interpunkce výstupu rozpoznávače pomocí automaticky vytvořených pravidel [48]. Tímto jsou promluvy rozděleny na věty. Čitelnost je dále zvýšena velkými písmeny na začátcích vět.

4.2 Databáze pro experimentální testování

V této práci je provedeno množství různých experimentů na různých datech. Data pro vyhodnocování výsledků spojité řeči jsou sdílena několika experimenty. Z časových důvodů není možné provést všechny experimenty na všech datech. V této kapitole jsou popsány řečové testovací databáze nejčastěji použité v této práci.

4.2.1 COST278

COST278 [49] je pan-evropská databáze anotovaných televizních nahrávek. Databázi tvoří 23 hodin nahrávek 10 televizních stanic v 7 evropských jazycích (holandština, portugalština, galština, čeština, slovinština, slovenština a řečtina). V této práci je pro účely vyhodnocování použita pouze česká část. Podrobnosti jsou uvedeny v tabulce 4.1.

Tabulka 4.1: Data COST278

COST278	Trénovací	Testovací
Počet řečových segmentů	498	339
Délka promluv	81 min	53 min
Počet slov	12922	8457
Počet různých slov	5446	3911

4.2.2 TV2005

Protože data COST278 jsou z roku 2003, byla vytvořena nová databáze obsahující nahrávky televizních zpráv z roku 2005. Podrobnosti jsou uvedeny v tabulce 4.2.

Tabulka 4.2: Data TV2005

TV2005	Testovací
Počet řečových segmentů	430
Délka promluv	66 min
Počet slov	9769
Počet různých slov	4312

4.3 Vyhodnocování výsledků rozpoznávání

Výsledky rozpoznávání izolovaných slov se udávají v procentech úspěšně rozpoznaných slov. Nechť C je počet správně rozpoznaných slov a N je celkový počet

slov, které byly rozpoznávány, pak přesnost (úspěšnost) rozpoznávání je dána

$$Acc = \frac{C}{N} \tag{4.1}$$

Vyhodnocení rozpoznávání spojité řeči je komplikovanější, protože je nutné určit, kolik slov bylo správně rozpoznáno h, kolik slov bylo vypuštěno (delece d) a kolik slov bylo přidáno (inzerce i). Referenční a rozpoznaný text jsou zarovnány tak, aby si správně rozpoznaná slova odpovídala, zbylá slova jsou označena jako substituce s, pokud je lze přiřadit jinému slovu, inzerce, pokud je slovo navíc v rozpoznaném textu, či delece, pokud chybí v rozpoznaném textu. Příklad zarovnání je uveden v tabulce 4.3. Více o zarovnávání je uvedeno v kapitole 8.

Tabulka 4.3: Zarovnávání referenčního a rozpoznaného textu

Reference Rozpoznaný text	Zítra	bude bude	oblačno opačně	a	až až	polojasno polojasno
Výsledek zarovnání	d	h	s	i	h	h

Úspěšnost rozpoznávání spojité řeči je definována vztahem

$$Acc = \frac{N - D - S\ - I}{N}, \tag{4.2}$$

kde N je počet slov v referenčním textu, D je počet delecí, I je počet inzercí a S je počet substitucí.

Další míra pro vyhodnocování úspěšnosti rozpoznávání je chybovost. Je definována vztahem

$$WER = 1 - Acc. \tag{4.3}$$

4.4 Test statistické významnosti

Neustálým zlepšováním různých částí automatického rozpoznávače mluvené řeči dochází ke snižování vlivu nových zlepšení na zvýšení úspěšnosti rozpoznávání. To je dáno tím, že je stále obtížnější zlepšit již velmi dobré jazykové a akustické modely. Aby bylo možné zjistit, zda nové vylepšení skutečně zvyšuje úspěšnost rozpoznávání, lze použít testy statistické významnosti. Tyto testy potvrzují, či vyvracejí hypotézu, zda zlepšení je statisticky významné. Praktická významnost nového vylepšení je většinou velmi obtížně měřitelná nebo ji lze zjistit pouze z dlouhodobého pozorování.

Pro test statistické významnosti zlepšení úspěšnosti rozpoznávání bylo Národním úřadem pro standardizaci a technologii [50] zavedeno několik testů statistické

významnosti. Nejvhodnějším testem statistické významnosti pro porovnávání výsledků spojitého rozpoznávače řeči je MAPSSWE test. Jde o běžný párový t-test o střední hodnotě normálního rozložení s neznámým rozptylem. Předpokladem pro správné použití testu je, aby byl proveden na alespoň 50 nezávislých náhodně vybraných promluvách [51].

Test zjišťuje počet chyb v promluvách porovnávaných výsledků. Jako chyba je bez rozdílu typu chápána delece, inzerce a substituce. Rozdíl počtu chyb v i-té promluvě je dán vztahem

$$Z_i = N_A - N_B, \tag{4.4}$$

kde N_A je počet chyb v promluvě A a N_B je počet chyb v promluvě B. Potom odhad střední hodnoty rozdílu chyb je

$$\hat{\mu}_Z = \sum_{i=1}^{n} \frac{Z_i}{n}, \tag{4.5}$$

Odhad rozptylu Z je

$$\hat{\sigma}_Z^2 = \frac{1}{n-1} \sum_{i=1}^{n} (Z_i - \mu_Z), \tag{4.6}$$

kde $\mu_Z = 0$. Pokud je použito k experimentu alespoň 50 promluv, pak náhodná veličina

$$W = \frac{\hat{\mu}_Z}{\frac{\hat{\sigma}_Z}{\sqrt{n}}} \tag{4.7}$$

má přibližně normální rozdělení. Nulovou hypotézu, $H_0 : \mu_Z = 0$, zamítáme na hladině významnosti $p\%$, pokud $|W| > N_{\frac{p}{2}}$, kde $N_{\frac{p}{2}}$ je $\frac{p}{2}$-kvantil normálního rozdělení. $\frac{p}{2}$ kvantil je použit, neboť je vyžadován oboustranný odhad μ_Z.

Tento test statistické významnosti je použit v práci pro testování statistické významnosti zlepšení výsledků spojitého rozpoznávání řeči, pokud je absolutní zvýšení úspěšnosti rozpoznávání malé.

Při testování hypotéz je zvolena hladina významnosti α, na které je zamítnuta nulová hypotéza. Hladina α udává pravděpodobnost chyby, které se dopustíme zamítnutím nulové hypotézy. Aby bylo zřejmé, jak daleko jsme od zvolené hladiny α, při zamítnutí či nezamítnutí nulové hypotézy, lze výsledek testu vyjádřit *p-hodnotou*, která udává hladinu významnosti, kterou je nutné použít pro testovaná data, aby byla nulová hypotéza zamítnuta na hranici zamítnutí. Pokud je tedy $\alpha \geq$ *p-hodnota*, pak dojde vždy k zamítnutí nulové hypotézy na hladině významnosti α.

Kapitola 5

Tvorba textového korpusu

Vytváření jazykového modelu a slovníku vyžaduje shromáždění velkého množství textu. Čím je větší slovník, tím více textu je potřeba. Nasbíraný text by měl být z oblasti, kde bude rozpoznávač řeči používán, aby slovník rozpoznávače pokryl co nejvíce nejčetnějších slov a jazykový model zachytil a spolehlivě odhadl hodnoty bigramů či trigramů. Je všeobecně známé, že často vyskytujících se slov je málo a řídce vyskytujících se slov je mnoho. To ukazuje i fakt, že přibližně polovina slov ze 2.4 milionu různých slov vyskytujících se ve 3.5 GB nasbíraných novinových článků byla viděna právě jednou.

5.1 Zdroje dat

Nejvíce veřejně přístupných textů se v dnešní době nalézá na webových stránkách. Pro přepis televizních a rozhlasových zpráv jsou nejvhodnější webové portály denního tisku, protože jsou snadno dostupné a nejvíce se obsahově přibližují vysílaným zprávám. Přepisy televizních a rozhlasových zpráv je nejběžnější dnešní aplikace rozpoznávačů spojité řeči kvůli vysoké kvalitě studiových nahrávek a snadnosti přístupu.

Pro automatizaci stahování webových stánek existuje množství nástrojů. Zrcadlo webového portálu lze jednoduše vytvořit například programem *wget*. Pokud je nutné další zpracování nasbíraných dat je výhodnější použít specializovanou knihovnu pro nějaký programovací jazyk, například knihovna *LWP* pro jazyk *Perl*, která je dobře popsána v [11]. *LWP* umožňuje transformaci webových stránek do stromové struktury a tím i snadné vyhledávání relevantních odkazů na další stránky. Stromovou strukturu lze také úspěšně použít k extrakci těch částí stránky, které obsahují užitečný text. Díky napojení *LWP* na programovací jazyk je možné třídit stažené stránky při procházení webového severu, což je výhodné zejména u portálů vytvářených dynamicky, kdy ze jména souboru, či cesty k němu není

zřejmý obsah. Setříděné texty se dají využít například k trénování identifikace článků.

Stahování webových stránek přináší i problémy. Získané články jsou většinou ukládány do velkého množství malých souborů. Běžní správci souborů mají problémy s velkým počtem položek v adresáři, proto je nutné dobré naplánovat strukturu adresářů, kam se stránky ukládají, aby bylo možné jednoduše uložené soubory prohlížet a kontrolovat. Další komplikace jsou na straně serveru, neboť jej systematické stahování stránek může neúměrně zatěžovat, což může vést až k jeho vyřazení z provozu a následnému zákazu přístupu na server od provozovatele. Program, který prohledává server, je omezen množstvím podmínek, které ošetřují výpadky spojení. Software na procházení portálu je často udělán „na míru", proto při změně designu stránek je nutné jej upravit. Programy na prohledávání internetu používají též internetové vyhledávací služby, například Google využívá program Geocrawler.

Jako zdroj textových dat pro přepisy televizních a rozhlasových zpráv lze využít služeb firem, které zprávy přepisují. Těchto dat je mnohem méně a jsou drahá, neboť se jedná o ruční přepisy. Na druhou stranu však nejlépe reflektují jazyk, který je používán ve zprávách.

Získat data pro lékařský diktovací systém je oproti novinovým článkům výrazně složitější a je jich mnohem méně. Lékařské zprávy často obsahují osobní údaje, které jsou chráněny zákonem proti zveřejnění. Osobní údaje není vždy možné spolehlivě eliminovat, proto nejsou tyto texty přístupné. Vzhledem k malému množství textů je možné spolehlivě odhadnout pravděpodobnosti výskytu unigramů, ne již bigramů. Lékařské diktovací systémy umožňují proto jen diktování izolovaných slov. Slovníky pro tyto systémy jsou menší.

5.2 Normalizace textového korpusu

Cílem normalizace získaných textů je přiblížení psané formy řeči formě vyslovované především expanzí zkratek a přepisem číslic. Normalizace se též snaží snížit počet různých zápisů slov, například „gymnázium, gymnasium" jsou stejná slova a zbytečně zvyšují velikost jazykového modelu a slovníku. Převodník těchto slov na unifikovanou formu byl vytvořen ručně [52]. Normalizace textu je prováděna ad-hoc. Normalizace korpusu pro rozpoznávač spojité řeči je prováděna v následujících krocích:

1. Expanze zkratek, které nejsou skloňovány, například: apod., tzn. Tyto jedno či dvouslovné zkratky jsou expandovány, neboť jsou expandovány i ve slovníku. Zkratky obsahující více slov jako s. r. o., v. o. s expandovány nejsou. Pokud by byly expandovány zkratky složené z několika slov, pak by se výrazně snížila čitelnost výstupu rozpoznávače, neboť expandovaná zkratka

není běžná a nikdo ji nečeká. Další důvod je, že tyto zkratky mají většinou i alternativní výslovnost. Například: s. r. o. se čte jako *eseró* nebo *společnost s ručením omezeným*.

2. Expanze číslic následovaných slovem letý, letou, . . . , např. 50letý na padesátiletý. Tento zkrácený zápis číslice je často používaný. Používá se však i expandovaná forma. Cílem expanze je sjednotit zápis těchto výrazů.

3. Přepis zkratky tzv. na takzvaná, takzvaný, . . . Jde o velmi četnou zkratku s netriviálním přepisem, neboť je její přepis skloňován. Přepis je možné provést s vysokou úspěšností v závislosti na sousedních slovech a možností přepisu není mnoho. Alternativou k expanzi této zkratky by byly alternativní výslovnosti ve slovníku. Tento přístup nebyl vyzkoušen, neboť rozpoznávač neumožňuje použít více jak 9 alternativních přepisů.

4. Standardizace, kdy jsou začátky vět označeny speciálním tagem. Tento krok je velmi používaný v mnoha rozpoznávačích. Začátek věty má svůj vlastní symbol. V některých publikacích je označován i konec věty. V této práci konec věty označován není, jedná se o duplicitní informaci k začátku věty. V tomto kroku jsou též všechny nealfanumerické znaky odděleny z obou stran mezerami, což je důležité pouze pro možnost aplikace dalších úprav, které na tuto skutečnost spoléhají.

5. Přepis zkratky hod. na hodin, hodina, . . . Cílem této úpravy je sjednotit zápis času a tím snížit počet alternativních variant. Přepis je netriviální, neboť slovo hodina je nutné skloňovat. Zkratka hod. se vyskytuje též ve fyzikálních jednotkách, například km/hod. Ne všechny zkratky jsou expandovány.

6. Přepis datumů [1], například *4. ledna* na *čtvrtého ledna*. Expanze číslic je nejproblematičtější částí normalizace korpusu, neboť jich je nekonečně mnoho a je často nutné je skloňovat. Proto je důležité odhadnout, co číslice znamenají. Přepis datumů lze provést jednodušeji, pokud víme, že jde o datum. Výskyt datumu lze jednoduše zjistit pomocí stavového automatu, neboť je jeho zápis ustálen. Přepisuje se zároveň číslice i ostatní části datumu, jejichž zápis je zkrácen.

7. Expanze číslic s předložkou s využitím znalosti mluvnických kategorií následujícího slova, například *ve 4. patře* na *ve čtvrtém patře*. Mluvnické kategorie jsou získány z morfologického analyzátoru [44]. Přeložka spolu s informací o rodě, čísle a pádě následujícího slova často vede k určení správného tvaru rozepsané číslovky. Tato metoda není úplně spolehlivá, ale ve

[1] blabla

většině případů použitelná. Pokud jsou pochyby o správném přepsání, není číslovka expandována. Pochyby vznikají, pokud je více různých možností, nebo předložka není kompatibilní se zjištěnými kategoriemi slova za číslicí.

8. Expanze číslic vyjadřujících čas ze spojení: V X hodin, například *v 5 hodin* na *v pět hodin*. Tato normalizace je speciální případ předchozí normalizace, ale do slovního spojení *22 hodin* se číslice přepisují jako by za nimi následovalo slovo v ženském rodě, po číslici 22 se přepisují, jako by za nimi následovalo slovo v mužském rodě.

9. Expanze zbylých číslic. Tato normalizace zahrnuje pravidla pro přepis desetinných čísel a číslovky částečně vyjádřené slovem, například *7 milionů* se přepíše na *sedm milionů*. Pokud je však před číslicí předložka, je aplikována úprava z bodu 7.

10. Aplikace převodníku jednoduchých slov. Pomocí ní se sníží počet alternativních textových variant slov. Například slova *benzín* a *benzin* znamenají stejnou věc a píší se podobně. Proto je podporována pouze jedna varianta. Preference varianty je většinou subjektivní, ale měla by být konzistentní s dalšími rody, pády či časy normalizovaného slova. Převodník je vytvořen manuálně. Nižší počet alternativních zápisů snižuje velikost slovníku.

11. Aplikace převodníku slov na spojování či rozdělování slov, například *apriori* na *a priori*. Tato úprava je obdoba předchozí úpravy. Z důvodů efektivnější implementace byly tyto úpravy rozděleny.

12. Expanze číslic před slovem krát, například *5krát* na *pětkrát*. Úprava postihuje častý výskyt těchto číslic po všech provedených expanzích číslic.

13. Drobné úpravy na základě vizuální inspekce korpusu. Pravidla úprav jsou vytvářena na základě vizuální inspekce korpusu a výsledků rozpoznávače. Tyto úpravy zahrnují například přepsání *př. n. l.* na *před naším letopočtem*.

Výsledný korpus je převeden na malá písmena.

5.2.1 Vliv normalizace na úspěšnost rozpoznávání

Pro zjištění vlivu normalizace korpusu na úspěšnost rozpoznávání byl proveden experiment, kdy byly originální texty postupně normalizovány. Z korpusů jednotlivých kroků normalizace byly vytvořeny jazykové modely. Experiment byl proveden na databázi TV2005 4.2.2. Výsledky jsou uvedeny v tabulce 5.1.

Z výsledků je patrné největší zvýšení úspěšnosti rozpoznávání při označování začátku vět a oddělení nealfanumerických znaků od slov. P-hodnota testu statistické významnosti zlepšení mezi originálními texty (pořadí operace 0) a poslední

Tabulka 5.1: Vliv normalizace na úspěšnost rozpoznávání

Název operace	pořadí operace	úspěšnost rozpoznávání (Acc)
Originální texty	0	78.64 %
Jednoduché zkratky	1	78.63 %
X-letý	2	78.50 %
Tzv.	3	78.65 %
Standardizace	4	79.90 %
Hod.	5	79.98 %
Datum	6	80.05 %
Číslice s předložkou	7	80.11 %
V X hodin	8	80.09 %
Ostatní číslice	9	80.03 %
Převodník	10	80.18 %
Převodník spojování slov	11	80.03 %
X krát	12	80.08 %
Ruční úpravy	13	80.04 %

úpravou (pořadí operace 13) je 4.6e-09. Textový korpus zahrnuje v současné době 3.5 GB textových souborů, převážně článků z denního tisku. Korpus obsahuje přibližně 519 milionů slov, z toho 2375859 různých.

5.3 Speciální úpravy lékařských textů

Diktování lékařských zpráv umožňuje zvýšit efektivitu práce lékařů například tím, že při diktování mohou držet v ruce rentgenové snímky, aniž by měli zaměstnány ruce psaním na klávesnici. Lékaři většinou nemají mnoho času, který by mohli věnovat konzultacím při vytváření diktovacího systému, proto je při normalizaci textu kladen důraz na použití jiných lékařských zdrojů. V této sekci je uveden postup čištění lékařských textů pro lékařský diktovací systém vyvíjený v Laboratoři počítačového zpracování řeči Technické univerzity v Liberci.

Typickým znakem dostupných lékařských textů, ze kterých má být vytvářen slovník a jazykový model, je maximální úspora psaní. Dokumenty obsahují veliké množství různých zkratek, z nichž některé jsou normalizovány, ostatní jsou specifické pro nemocnici či oddělení. Lékaři používají více lokalizovaných forem latinských výrazů (latinský základ a česká přípona), čímž je ztížena automatická fonetická transkripce, neboť tato slova mají fonetický přepis odlišný od běžných českých zvyklostí. Překlepy jsou též běžnou součástí lékařských zpráv. Na základě prohlídky několika textů se normalizace lékařských textů soustředila na opravu překlepů, expanzi zkratek, výběr slov do slovníku a identifikaci slov s neobvyk-

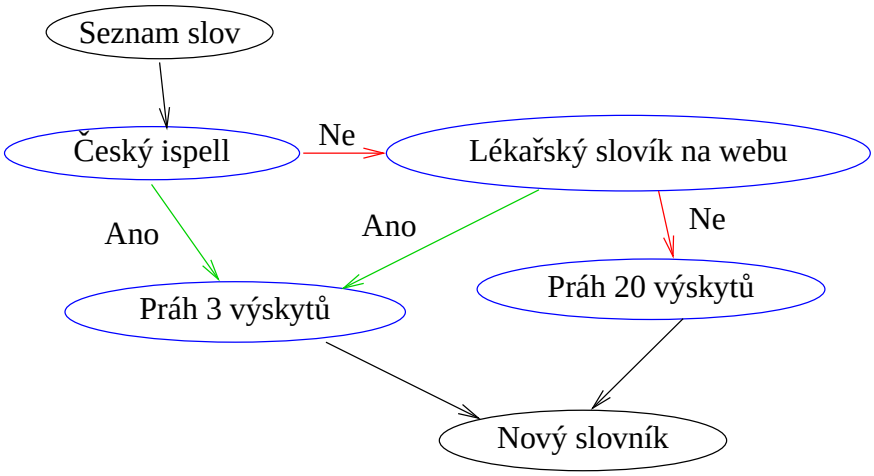
lým fonetickým přepisem. Dostupných lékařských textů je většinou málo na to, aby se překlepy eliminovaly svojí malou četností výskytu. Proto je nutné úpravám textů věnovat větší pozornost.

5.3.1 Oprava překlepů a expanze zkratek

Většina překlepů a zkratek může být opravena pouze ručně, protože jen specialisté vědí, zda slovo je překlep a které zkratky jsou vyslovovány tak, jak se píší a které mají být expandovány. Pouze nejčetnější překlepy a zkratky jsou opraveny a expandovány, protože specialisté mají málo času. Některé korekce překlepů a expanze zkratek byly převzaty z [53].

5.3.2 Výběr slov do slovníku

Dostupné texty s expandovanými zkratkami a opravenými nejčetnějšími překlepy stále obsahují mnoho chyb, proto je seznam slov vytvořený z textů zpracován českou automatickou kontrolou pravopisu poskytovanou programem ispell a porovnán s lékařským slovníkem dle obrázku 5.1.



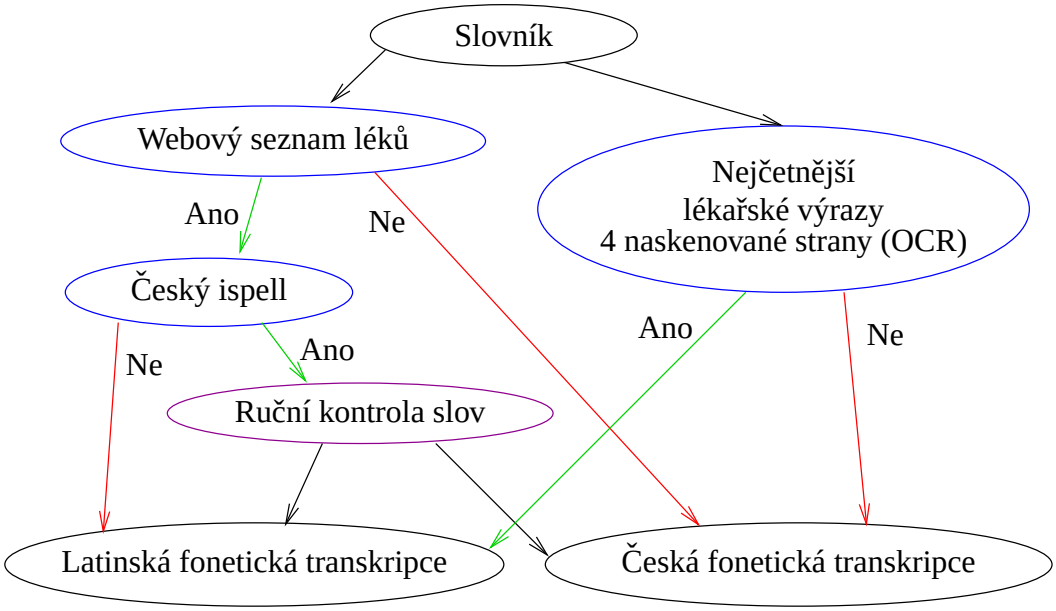
Obrázek 5.1: Výběr slov do lékařského slovníku

Pokud kontrola pravopisu zná slovo, a to se vyskytuje v textech častěji než 3krát, je zařazeno do nového slovníku. Práh je použit, protože ispell propouští některé překlepy. Pro slova, která neprojdou kontrolou pravopisu je zjišťováno, zda existují v lékařském slovníku dostupném jako webová aplikace [54]. Pokud je slovo v lékařském slovníku, nebo slovník nabídne podobné slovo, je zařazeno do nového slovníku, pokud se vyskytuje alespoň 3krát. Pro slova, která nejsou

uvedena ani v lékařském slova, je zvolen práh výskytu 20, aby mohla být zařazena do slovníku. Může se totiž jednat o jinou formu zápisu latinského slova, či odvozeného slova. Ověřování, zda je slovo v lékařském slovníku je provedeno automaticky jednoduchou aplikací s použitím balíku LWP pro Perl.

5.3.3 Identifikace slov s latinským fonetickým přepisem

Latinská slova je možné přepsat podle pravidel uvedených v knize [55]. Nejprve je ovšem nutné identifikovat slova, na která se aplikuje latinský fonetický přepis. Kandidátmi na latinský přepis jsou názvy léků a nejběžnější latinská slova z lékařské literatury. Z knihy [55] byly naskenovány 4 strany nejběžnějších lékařských termínů a programem FineReader převedeny na text. Zda je slovo názvem léku, bylo možno provést automaticky ze stránek pojišťovny [56] podobně jako v sekci 5.3.2, kde se ověřuje, zda je slovo v lékařském slovníku. Postup identifikace slova s latinskou fonetickou transkripcí je uveden na obrázku 5.2.



Obrázek 5.2: Výběr slov s latinským fonetickým přepisem

Slovo ze slovníku je porovnáváno buď se seznamem nejfrekventovanějších lékařských výrazů nebo se seznamem léků. Pokud není uvedeno ani v jednom ze seznamů, je provedena automatická česká fonetická transkripce. Na slova vyskytující se v seznamu nejfrekventovanějších lékařských výrazů je aplikována latinská fonetická transkripce. Pokud je slovo obsaženo v seznamu léků a zároveň není akceptováno českým ispellem, je přepsáno dle latinských fonetických pravidel. V případě, že je slovo akceptováno českým ispellem, pak jde buď o české

slovo v názvu léku, nebo velmi frekventovaný lék. Takových slov bylo přibližně 40, proto nebyl problém je ručně zkontrolovat.

5.4 Zhodnocení

Cílem kapitoly je ukázat několik postupů normalizace textu a vliv normalizace na úspěšnost rozpoznávání. Nejprve je ukázána normalizace běžných textů. Další sekce se zabývají odlišným příkladem, kdy je třeba vyčistit lékařské texty a vybrat slova do slovníku s co nejmenšími nároky na čas lékařů strávený na vývoji lékařského diktovacího systému. V tomto případě je nutné efektivně využít co nejvíce dostupných zdrojů–online slovníky, seznamy léků, automatickou kontrolu pravopisu a automatické rozpoznávání znaků (OCR). Lékařských textů je mnohem méně a jsou více „znečištěné” než novinové články.

Kapitola 6

Tvorba slovníku

6.1 Principy výběru slov do slovníku

Slovník je důležitou součástí rozpoznávače řeči, neboť rozpoznávač pracuje pouze se slovy ve slovníku. Pokud není slovo ve slovníku, nemůže být rozpoznáno. Slovník je vytvářen z velkého množství textu vybráním nejčetnějších slov, tím se dosáhne vysokého pokrytí zdrojového textu slovníkem. Text, ze kterého je vytvářen slovník, musí obsahovat témata, pro která je rozpoznávač navrhován. Pro požadované pokrytí je velikost slovníku závislá na jazyku, pro který je slovník vytvářen. Angličtina obsahuje málo slov, proto stačí slovník o desítkách tisíc slov. Mnoho slovních tvarů jazyka zvyšuje velikost slovníku. Aby se dosáhlo stejného pokrytí českého textu jako anglického, je nutné použít větší slovník. Pro 99% pokrytí anglického či španělského textu je třeba slovník o velikosti 65 tisíc slov [57], [58].

Velké slovníky jsou problematické, neboť vyžadují velké korpusy, jazykový model je pak také veliký a doba rozpoznávání se značně prodlužuje. Velký slovník je též náročný na údržbu. Témata promluv se v čase mění, a je tudíž nutné přidávat aktuální slova, případně odebírat slova zastaralá. S přibývající velikostí slovníku je mezi kandidáty na přidání do slovníku stále více nesmyslných slov a překlepů, neboť mohou být četnější, než zřídka se vyskytující správná slova. Z tohoto důvodu není udržování a adaptace slovníku plně automatizovatelná. Efektivní správě slovníku významně pomáhá vhodný software. Tento software musí kontrolovat správný tvar slovníku, který očekává rozpoznávač (například správné setřídění). Dále je nutné kontrolovat, zda fonetický přepis obsahuje pouze povolené znaky odpovídající modelům fonémů, zda se stejné slovo nevyskytuje vícekrát nebo jestli již neexistuje tvar, na který je slovo normalizováno převodníkem z předchozí kapitoly. Další podpora efektivity správy slovníku spočívá v možnosti distribuce správy mezi více lidí, je tedy nutno pracovat s více verzemi slovníku.

Příklad slovníku je uveden v tabulce 6.1.

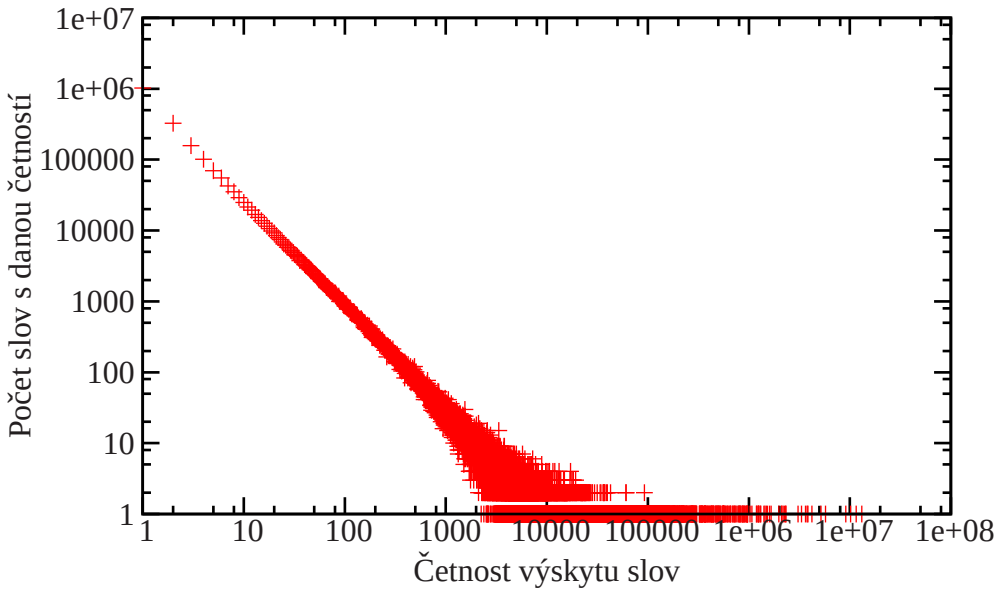
Tabulka 6.1: Příklad slovníku

Slovo	výslovnosti
absolventi	apsolveňťi
absolventka	apsolventka
absolventkami	apsolventkami
absolventkou	apsolventkou
absolventku	apsolventku
absolventky	apsolventki
absolventovi	apsolventovi
absolventskou	apsolvenckou
absolventská	apsolvencká
absolventské	apsolvencké
absolventského	apsolvenckého
absolventském	apsolvenckém
absolventský	apsolvenckí
absolventských	apsolvenckíX apsolvenckíh
účinkujícím	účiNkujícím
účinkujícími	účiNkujícími
účinky	účiNki
účinků	účiNkú
účinkům	účiNkúm
účinnost	účinost účinozd
účinnosti	účinosťi
účinností	účinosťí
účinnou	účinou

6.2 Charakteristiky slovníku pro rozpoznávač řeči

Slovník pro rozpoznávání řeči odvozen z textového korpusu obsahujícího 3.5 GB textu, převážně článků z denního tisku. Korpus obsahuje přibližně 519 milionů slov, z toho 2375859 různých. Histogram četností slov v textovém korpusu, viz obrázek 6.1, ukazuje, že četných slov je jen málo, zatímco různých slov s občasným výskytem je mnoho, což je důvod používání velkých slovníků pro rozpoznávání češtiny.

Pokrytí textového korpusu různě velikými slovníky je uvedeno v tabulce 6.2. Ze slovníku používaném v rozpoznávači spojité řeči obsahujícím 312 tisíc slov



Obrázek 6.1: Histogram četností výskytu slov v textovém korpusu. Graf je vykreslen v logaritmických souřadnicích.

Tabulka 6.2: Pokrytí textového korpusu různě velikými slovníky

Počet slov ve slovníku	Počet pokrytých slov	Pokrytí
25000	447807165	86.23 %
50000	474605302	91.39 %
75000	486883723	93.76 %
100000	494064808	95.14 %
125000	498796835	96.05 %
150000	502145401	96.70 %
175000	504637124	97.18 %
200000	506561303	97.55 %
225000	507514229	97.73 %
250000	509313731	98.08 %
275000	510327772	98.27 %
300000	511175175	98.44 %

byly odvozeny menší slovníky na základě četnosti výskytů slov v textovém korpusu.

6.3 Fonetická transkripce

Při rozpoznávání řeči je důležité vzájemné spojení textové a akustické formy slova. Pro každé slovo je nutné mít jeho akustický model, který je porovnáván s akustickým signálem přicházejícím do mikrofonu. Pro velké slovníky není možné vytvořit akustický model pro každé slovo zvlášť. Proto jsou vytvářeny akustické modely pro menší stavební jednotky slova, například fonémy, a ty jsou poté spojovány. Foném je nejmenší jednotka řeči, která může rozlišovat jednotlivá slova [9]. Fonémů je podstatně méně než slov. Je možné používat i jiné větší stavební jednotky slov, ale vždy je nutné volit kompromis mezi počtem jednotek, složitostí přepisu textu na jednotky a dostatečným množstvím dat, ze kterého jsou akustické modely natrénovány.

Pro zápis fonémů je možno použít mezinárodní fonetickou abecedu (IPA) popsanou v [20]. Pro češtinu byla vypracována abeceda PAC [21] přehledněji vystihující česká fonetická pravidla. Česká fonetická abeceda je i implementačně výhodnější. Česká fonetická abeceda je uvedena v tabulce 6.3. Rozpoznávače řeči používané v této práci pracují s modely fonémů, kterých je 40 [21]. Dále jsou přidány modely nejběžnějších šumů a hluků [47].

Fonetická transkripce je přepis textové podoby slova na sekvenci fonémů. V každém jazyce existují x fonologická pravidla jak provádět fonetickou transkripci (vyslovovat slova). V některých jazycích, jako je angličtina existuje veliké množství pravidel. Naopak v češtině nebo němčině je pravidel mnohem méně a je možné je jednodušeji implementovat.

Fonetická transkripce češtiny není jen pouhý přepis písmen na odpovídající fonémy podle tabulky uvedené v [21]. Často dochází ke koartikulaci, kdy je písmeno přepsáno na foném v závislosti na jeho okolí. V tomto případě může být jedno písmeno přepsáno na několik různých fonémů nebo úplně vypuštěno z přepisu.

Fonologická pravidla mohou být implementována ve formě produkčních pravidel [22], [23], rozhodovacích stromů [24], konečných automatů [25] nebo neuronové sítě [26, 59].

6.3.1 Fonologická pravidla

Pro češtinu bylo fonetiky vypracováno množství fonologických pravidel [9] přepisujících hlásky na fonémy. České hlásky jsou rozděleny na samohlásky a souhlásky a souhlásky jsou rozděleny na znělé a neznělé. Rozdělení je uvedeno v tabulce 6.4. Fonologická pravidla jsou ve formě produkčních pravidel a popisují jak přepsat hlásky na fonémy v závislosti na jejich okolí. Pravidla zahrnují i pokročilé jevy jako je například spodoba znělosti a spodoba artikulační. Následují

Tabulka 6.3: Česká fonetická abeceda (PAC)

zápis fonému dle PAC	foném vyjádřený českými hláskami	příklad (fonetický přepis)
a	a	tát**a** (tát**a**)
á	á	t**á**ta (t**á**ta)
b	b	**b**ota (**b**ota)
c	c	o**c**el (o**c**el)
C	dz	**Dz**urinda (**C**urinda)
č	č	**č**ivava (**č**ivava)
Č	dž	Al**Dž**azíra (al**Č**azíra)
d	d	je**d**en (je**d**en)
ď	ď	**dě**lat (**ď**elat)
e	e	l**e**v (l**e**v)
é	é	m**é**ně (m**é**ňe)
f	f	**f**auna (**f**auna)
g	g	**g**uma (**g**uma)
h	h	a**h**a (a**h**a)
X	ch	**ch**udoba (**X**udoba)
i	i nebo y	v**y**p**i**l (v**i**p**i**l)
í	í nebo ý	v**ý**pis (v**í**pis)
j	j	do**j**at (do**j**at)
k	k	**k**olo (**k**olo)
l	l	ko**l**o (ko**l**o)
m	m	**m**oje (**m**oje)
M	nosové m	tra**m**vaj (tra**M**vaj)
n	n	**n**os (**n**os)
N	nosové n	ba**n**ka (ba**N**ka)
ň	ň	**ně**co (**ň**eco)
o	o	v**o**da (v**o**da)
ó	ó	**ó**da (**ó**da)
p	p	**p**rase (**p**rase)
r	r	p**r**ase (p**r**ase)
ř	ř	mo**ř**e (mo**ř**e)
Ř	znělé ř	ke**ř** (ke**Ř**)
s	s	**s**eno (**s**eno)
š	š	**š**um (**š**um)
t	t	du**t**ý (du**t**í)
ť	ť	**tí**seň (**ť**íseň)
u	u	d**u**še (d**u**še)
ú	ú nebo ů	r**ů**že (r**ú**že)
v	v	**v**etřelec (**v**etřelec)
z	z	ko**z**a (ko**z**a)
ž	ž	**ž**ně (**ž**ňe)

Tabulka 6.4: Znělost českých hlásek

Samohlásky (SA)	a, á, e, é, ě, i, í, y, ý, o, ó, u, ú, ů
Znělé párové souhlásky (ZPS)	b, d, ď, g, z, ž, v, h, dz(C), dž(Č)
Neznělé párové souhlásky (NPS)	p, t, ť, k, s, š, f, ch(X), c, č
Jedinečné souhlásky znělé (JS)	m, n, ň, l, j, r, ř

příklady pravidel:

Následuje-li *ě* po *b, p, f, v* přepisuje se na *je*
ě → je / <b, p, f, v>_
Jestliže *x* stojí před znělou souhláskou, přepisuje se na *gz*, jestliže stojí před neznělou souhláskou či na konci slova, přepisuje se na *ks*.
x → gz /_<ZPS, JS>
x → ks /_<NPS, ->

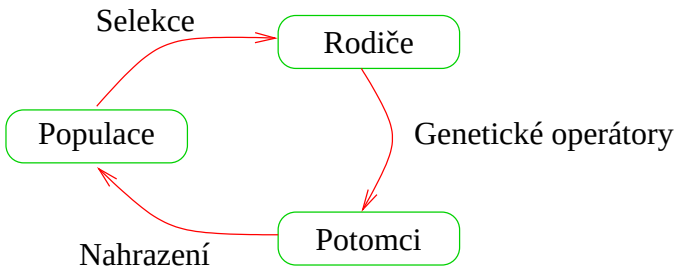
Slovník obsahuje pouze jednotlivá slova, proto pravidla pro spodobu znělosti mezi slovy jsou používána jen výjimečně jako alternativní výslovnost slova. Česká fonologická pravidla fungují spolehlivě pro česká slova. Cizí slova, zejména ta, kde se vyskytují slabiky *di, ti, ni* bývají často špatně přepisována na *ď'i, ť'i, ňi*. Například slovo *antimon* je přepsáno podle českých pravidel na *ant'imon* nikoli na *antymon*. Některá slova jsou přepisována podle českých i cizích pravidel současně, například slovo *antirasisti* má být přepsáno na *antyrasisť'i*, ale podle českých fonologických pravidel by bylo přepsáno na *ant'irasisť'i*. V příkladech je použit foném *y*, přestože není v PAC. Cílem je čitelnost příkladů. Ve skutečné fonetické transkripci není rozdíl mezi *y* a *i*, proto se obě zapisují jako *i*

Řešení správného přepisu cizích slov spočívá v zavedení výjimek. Výjimky se při fonetické transkripci aplikují jako první. Standardní fonologická pravidla jsou aplikována jako druhá. Výjimek však rychle přibývá a stávají se nepřehlednými, což může vést k poškození správné fonetické transkripce slov, která byla správně přepsána standardními fonologickými pravidly.

Jiným řešením fonetického přepisu cizích slov je odvození nových pravidel ze známých přepisů slov. Tím se i sníží počet výjimek. V této práci jsou odvozována nová fonologická pravidla ve formě produkčních pravidel. Originální sada pravidel je převzata z [22]. Nová pravidla mají původní sadu rozšířit, proto jsou ve stejné formě. Zabrání se tím reimplementaci původních pravidel. Nová fonologická pravidla jsou odvozována pomocí gramatické evoluce přímo do požadovaného formátu. Odvozování nových fonologických pravidel je uvedeno v [10].

6.3.2 Gramatická evoluce

Gramatická evoluce je evoluční algoritmus pomocí kterého lze vyvinout program popsatelný bezkontextovou gramatikou, který co nejlépe splňuje zadaná kritéria. Více o gramatické evoluci (GE) lze nalézt v [60]. Gramatická evoluce stejně jako ostatní evoluční algoritmy používá evoluční cyklus uvedený na obrázku 6.2. Jedinci jsou reprezentováni binárním řetězcem. Každý jedinec reprezentuje jeden



Obrázek 6.2: Evoluční cyklus

kompletní program. Generování syntakticky správných programů je zabezpečeno vhodně navrženou bezkontextovou gramatikou společnou pro všechny jedince. Použitá pravidla bezkontextové gramatiky, která generují konkrétní program, jsou dána právě binárním řetězcem (genotypem) jedince. Noví jedinci jsou během evoluce tvořeni genetickými operátory jako je mutace a křížení, aplikovanými na genotypy rodičů. Po skončení evoluce je vybrán nejlepší jedinec reprezentující nejlepší program ve smyslu zadaného kritéria neboli fitness.

6.3.3 Nová fonologická pravidla

Jak již bylo uvedeno, jsou fonologická pravidla odvozována jako produkční pravidla ve formátu

$$\text{písmeno} \rightarrow \text{foném}/\text{prefix}\ \_\text{postfix},\ \text{krok}, \tag{6.1}$$

kde prefix a postfix jsou sekvence písmen předcházející a následující přepisované písmeno. Krok označuje, kolik následujících písmen má být při fonetické transkripci přeskočeno. Krok umožňuje přepsat několik písmen najednou. Při odvozování nových fonologických pravidel jsou počet kroků, foném a přepisované písmeno fixní. Tím je zjednodušeno učení nových pravidel a zároveň je tak možné se soustředit na případy, které jsou nejčastěji chybně přepsány jako jsou slabiky di, ti, ni.

Nová pravidla by měla pokrýt co nejvíce špatně přepsaných slov a zároveň by neměla bránit aplikaci původních pravidel, pokud je jimi slovo správně přepisováno. Fonetický přepis je prováděn v následujících krocích.

1. Aplikuj výjimky.

2. Aplikuj nová pravidla.

3. Aplikuj originální pravidla.

Pravidla jsou uspořádána od nejspecifičtějších aplikovatelných na málo případů po nejobecnější aplikovatelná na libovolné písmeno bez ohledu na jeho kontext. Pokud je nějaké pravidlo aplikováno, je písmeno přepsáno a další pravidla se již na něj neaplikují. V opačném je na písmeno aplikováno následující obecnější pravidlo.

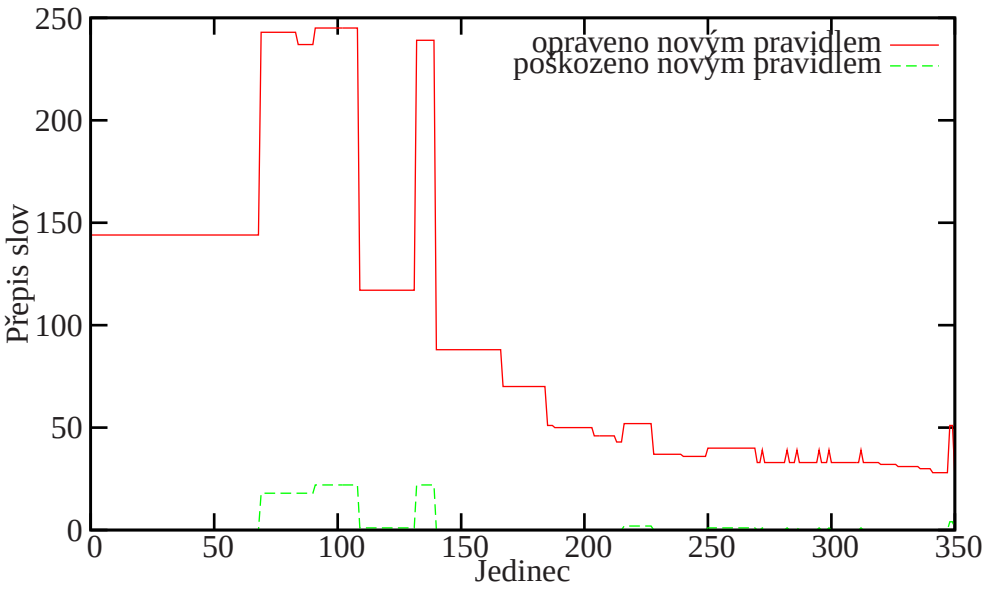
Při odvozování nových fonologických pravidel jsou výjimky ignorovány. Učené pravidlo je aplikováno jako první. Následně jsou aplikována ostatní pravidla.

6.3.4 Trénovací a testovací data

Trénovací a testovací množiny jsou vytvořeny pro každou trojici {písmeno, foném, krok} zvlášť. Hledá se jen prefix a postfix. Trénovací a testovací vzorky jsou vybrány ze slovníku obsahujícího 200000 slov. Všechna slova obsahující přepisované písmeno jsou vybrána ze slovníku a roztříděna do třech skupin. První skupina obsahuje slova, která jsou správně přepsána pomocí originálních fonologických pravidel. Druhá skupina je tvořena slovy, která by mohla být správně přepsána, kdyby bylo aplikováno nějaké nové pravidlo, které je hledáno. Třetí skupina jsou slova, u nichž nelze jednoduše odhadnout, zda jejich přepis nové pravidlo opraví. Toto rozdělení lze provést plně automaticky tak, že se pro všechna slova špatně přepsaná originálními pravidly provede i alternativní přepis hledaným pravidlem s prázdným prefixem i postfixem. Pokud je mezi alternativami přepis shodný s přepisem ze slovníku, pak je možné fonetický přepis opravit nějakým novým pravidlem. Trénovací a testovací množiny jsou vytvořeny z prvních dvou skupin a to tak, že dvě třetiny jsou trénovací a jedna třetina testovací.

6.3.5 Experimenty a výsledky

Všechny experimenty probíhaly s populací tvořenou 500 jedinci. Během evoluce bylo vyhodnoceno 50500 jedinců. Rodiče, na které byly aplikovány genetické operátory byli vybíráni turnajovou selekcí mezi třemi jedinci. Nová populace byla vytvářena metodou „steady state“ [61] tak, že 80 % jedinců zůstalo a 80 % nových jedinců bylo vytvořeno křížením, ostatní mutací. Diverzita populace byla udržována metodou LICE [62].



Obrázek 6.3: 350 nejlepších jedinců poslední populace

Fitness funkce neboli kritérium výběru jedince je popsáno níže. Trénovací množina je rozdělena na slova správně přepsaná originálními fonologickými pravidly C a slova, jejichž automatická transkripce může být opravena novým pravidlem. Nechť B je počet slov s transkripcí opravenou novým pravidlem a $\bar{C}$ je počet pravidel z C, jejichž přepis je novým pravidlem poškozen. Fitness je pak

$$f = B - w\bar{C}, \tag{6.2}$$

kde $w \in \langle 0, \infty \rangle$ je váha penalizující transkripci poškozenou novým pravidlem. Pro všechny experimenty je $w = 6$.

V rámci experimentů jsou hledána pravidla pro nejproblematičtější slabiky di, ti, ni. Výsledná fonologická pravidla byla ručně vybírána z poslední populace, a to taková pro která bylo $\bar{C} = 0$, tedy žádný přepis nebyl novým pravidlem poškozen. 350 nejlepších jedinců poslední populace je ukázáno na obrázku 6.3

Bylo nalezeno 38 nových pravidel, která byla přidána k 248 originálním pravidlům.

Slovník s 200000 slovy byl přepsán pomocí originálních fonologických pravidel s výjimkami a s novou sadou fonologických pravidel bez použití výjimek. Úspěšnost přepisu je uvedena v tabulce 6.5. Počet slov, která jsou opravena jednotlivými pravidly není lehké přesně určit, neboť slovo může být opraveno dvěma pravidly zároveň.

Počty opravitelných a opravených fonetických přepisů jsou uvedeny v tabulce 6.6.

Tabulka 6.5: Experimentální výsledky s novými fonologickými pravidly

	Správně přepsáno	úspěšnost
Originální pravidla s výjimkami	185237	93 %
Přidána nová pravidla	189807	95 %

Tabulka 6.6: Opravitelné a opravené chyby fonetické transkripce

	di	ti	ni	celkem
Opravitelné chyby	3746	1392	2021	7159
Opravené chyby	-	-	-	4570

Vyhodnocení

Výsledky ukazují, jak lze téměř automaticky vylepšit fonetickou transkripci. Uvedený přístup znovu neobjevuje všeobecně známá fonologická pravidla. Gramatická evoluce umožnila najít pravidla v takovém formátu, aby je bylo možné použít v existujícím systému automatické fonetické transkripce. Jak bylo předpokládáno, nová nalezená pravidla jsou velmi specifická a aplikovatelná na menší počet slov než originální fonologická pravidla.

6.4 Slovní spojení ve slovníku

Na základě analýzy rozpoznávaných promluv bylo zjištěno, že krátká slova jsou často špatně rozpoznána. Metody pro podrobnější analýzu výsledků rozpoznávače jsou uvedeny v kapitole 8. Krátká slova jsou ignorována, rozpoznána jako šum, nebo přidána jako předpona či přípona následujícího nebo předcházejícího slova. Dlouhá slova jsou většinou rozpoznána správně. Slovní spojení krátkého frekventovaného slova a jeho častého následníka, či předchůdce může zvýšit úspěšnost rozpoznávání, neboť je toto spojení chápáno rozpoznávačem jako jedno dlouhé slovo. Slovní spojení jsou již do slovníku rozpoznávačů spojité řeči přidávána [12, 63] ručně. Cílem této sekce je zjistit vliv plně automatického přidávání slovních spojení do slovníku na úspěšnost rozpoznávání.

Dalším důvodem pro přidávání slovních spojení je fakt, že slovní spojení „lokálně zvýší" řád jazykového n-gramového modelu. Pokud přidáme do slovníku spojení *v_sobotu*, pak *v_sobotu večer* je již trigram. Velké slovníky, 200 tisíc slov a více, způsobují, že rozpoznávače řeči potřebují i velké jazykové modely. Výpočet českého bigramového modelu pro slovník s 312 tisíci slovy spotřebuje 0.9 až 1.5 GB RAM v závislosti na požadavcích na rychlost výpočtu. Spolehlivý odhad podmíněných pravděpodobností takového modelu vyžaduje také veliké množství textu. Proto je plně trigramový jazykový model pro veliké slovníky obtížně reali-

zovatelný.

Třetí důvod pro přidání slovních spojení je, že spojení řeší problematiku koartikulace, kdy je slovo vyslovováno různě v závislosti na kontextu okolních slov. Tento problém je také řešen přidáním různých výslovnostních variant slova [64, 65].

Přirozená slovní spojení, která se v některých jazycích běžně vyskytují, mohou způsobovat problémy tím, že zvětšují velikost slovníku a zvyšují řídkost textového korpusu, ze kterého je počítán jazykový model. Tyto problémy se týkají zejména jazyků, kde se nová slova běžně vytvářejí spojením existujících slov jako například v němčině nebo finštině. Několik postupů, jak rozbíjet tato spojení bylo publikováno v [66], [67].

Slovní spojení automaticky přidávaná do slovníku jsou tvořena ze slov již ve slovníku existujících, čímž je eliminováno riziko vložení překlepu, či nesmyslného slova. Slovní spojení mohou být vybírána buď na základě vzájemné informace, nebo četnosti výskytu spojení v textovém korpusu.

6.4.1 Míry pro výběr slovních spojení

Kritérium pro výběr vhodného slovního páru musí splňovat následující požadavky:

- Slovní spojení musí obsahovat alespoň jedno krátké slovo. Slovní spojení dlouhých slov přispívá pouze k řídkosti textového korpusu. Dlouhá slova nejsou cílem optimalizace rozpoznávače.
- Slovní spojení musí být četné, aby se pouze nezvětšoval slovník a řídkost dat.
- Slova ve slovním spojení musí být četná, neboť četná slova jsou spíše česká než cizí a je možné aplikovat automatický fonetický přepis s nižším rizikem nesprávného přepisu.

Jako krátká slova jsou chápána slova mající maximálně 3 znaky a minimální četnost výskytu každého slovního spojení je stanovena na 30.

Vzájemná informace

Vzájemná informace je často používána k výběru kolokací. Kolokace jsou slova, která se často vyskytují spolu a zřídka zvlášť. Vzájemná informace je definována následovně:

$$PMI = \log\left(\frac{p(w_1, w_2)}{p(w_1)p(w_2)}\right), \tag{6.3}$$

kde $p(w_1, w_2)$ je pravděpodobnost sekvence slov w_1 a w_2, $p(w_1)$ je pravděpodobnost slova w_1 jako předchůdce a $p(w_2)$ je pravděpodobnost slova w_2 jako následníka.

Četnost výskytu slovního spojení

Četnost výskytu slovního spojení je nejjednodušší způsob výběru slovního spojení. Četnost výskytu splňuje požadované vlastnosti na kriteriální funkci a je počítána při vytváření jazykového modelu.

6.4.2 Přidávání slovních spojení do slovníku

Slovní spojení jsou do slovníku přidána jako samostatná slova, přičemž jednotlivá slova slovního spojení jsou oddělena znakem '_'. Tento znak je odstraněn z výstupu rozpoznávače. Fonetická transkripce slovního spojení je provedena plně automaticky pomocí fonologických pravidel, přičemž slovní spojení je transkripci předloženo jako jediné slovo.

Slovní spojení musí být také vložena do textového korpusu a jazykový model musí být následně z tohoto korpusu znova spočítán.

6.4.3 Experimenty

Experimenty byly prováděny na databázi COST278, viz část 4.2.1. Základní úspěšnost rozpoznávání pro slovník bez slovních spojení bylo 74.48 %. Základní úspěšnost rozpoznávání se slovníkem s manuálně vybranými 1731 slovními spojeními bylo 75.80 %. P-hodnota (část 4.4) při testování s ručně vybranými spojeními oproti slovníku bez spojení je 1.1e-04. Při manuálním výběru slovních spojení byl brán ohled na: kolokace, slova objevující se často spolu a zřídka zvlášť, běžná spojení předložek a následujícího slova a časté slovní páry s nestandardní fonetickou transkripcí.

Slovní spojení s nejvyšší hodnotou vzájemné informace (PMI) a s nejvyšší četností výskytu byla přidána do slovníku. Výsledky jsou uvedeny v tabulce 6.7.

Výběr slovních párů pomocí PMI nepřinesl zlepšení v úspěšnosti rozpoznávání. To je způsobeno tím, že kolokace vybrané pomocí PMI nejsou dostatečně četné. Četnost výskytu se ukázala být vhodnějším kritériem pro výběr slovních párů. Přidání 10000 slovních párů zlepšilo úspěšnost téměř neznatelně. Zlepšení není statisticky významné na hladině významnosti 5 %.

Mnoho vybraných spojení obsahuje různé předložkové vazby. Předložkové vazby mohou také způsobovat koartikulaci. V dalším experimentu jsou slovní spojení znovu vybírána na základě četnosti, ale předložky musí být pouze na prvním

Tabulka 6.7: Výsledky rozpoznávání se slovními spojeními vybranými na základě vzájemné informace PMI a četnosti výskytu.

Přidaných spojení	úspěšnost rozpoznávání (Acc)	
	PMI	četnost výskytu
1000	74.59	75.40
2000	74.55	75.73
3000	74.55	75.95
4000	74.50	76.20
5000	74.64	76.05
6000	74.70	75.89
7000	74.67	75.78
8000	74.68	76.04
9000	74.64	76.28
10000	74.60	76.33

místě slovního spojení. Z výsledků uvedených v tabulce 6.8 je patrné, že k výraznému zlepšení nedošlo, což je způsobeno tím, že většina předložek ve slovních spojeních vybraných na základě četnosti již na prvním místě je.

Tabulka 6.8: Výsledky rozpoznávání se slovními spojeními vybranými na základě četnosti výskytu, přičemž předložka může být pouze na prvním místě slovního spojení.

Přidaných spojení	úspěšnost rozpoznávání (Acc)	
	četnost výskytu	četnost výskytu s předložkou na 1. místě
1000	75.40	75.37
2000	75.73	76.15
3000	75.95	76.11
4000	76.20	76.11
5000	76.05	76.52
6000	75.89	76.59
7000	75.78	76.81
8000	76.04	76.78
9000	76.28	76.89
10000	76.33	76.99

Výsledky předcházejících experimentů ukazují, že přidávání slovních spojení na základě četnosti výskytu zvyšuje úspěšnost rozpoznávání. Z tabulky 6.8 je patrné mírné zvýšení úspěšnosti rozpoznávání s rostoucím počtem přidaných slovních spojení. Přidání 10000 slovních párů s předložkou na začátku již zvýšilo

úspěšnost rozpoznávání oproti slovníku s ručně přidanými slovními spojeními, což je 75.8 %. Zlepšení je statisticky významné, p–hodnota je 8.4e-04.

Následující experimenty ukazují případy, kdy je slovních spojení přidáno více. Tabulka 6.9 ukazuje případ, kdy je přidáno více slovních spojení. Stagnace a mírné

Tabulka 6.9: Více slovních spojení přidaných na základě četnosti výskytu.

Přidaných spojení	úspěšnost rozpoznávání (Acc)	
	četnost výskytu	četnost výskytu s předložkou na 1. místě
10000	76.33	76.99
15000	76.82	77.68
20000	77.13	77.57
25000	77.37	77.65
30000	77.43	77.77
35000	77.57	77.88
40000	77.43	77.90
45000	77.69	77.94
50000	77.46	77.91
55000	77.45	77.90

snižování úspěšnosti rozpoznávání je patrné od 45000 přidaných slovních spojení. Pro 45000 přidaných slov je p-hodnota rovna 4.0e-07 pro zamítnutí hypotézy o stejných výsledcích jako pro slovník s ručně přidanými slovními spojeními.

Kompletní tabulka se všemi provedenými experimenty je v příloze B.

6.4.4 Analýza výstupu rozpoznávače

Cílem přidávání slovních spojení bylo eliminovat chyby při rozpoznávání krátkých slov. Tabulka 6.10 ukazuje nejčastější chyby rozpoznávače bez slovních spojení a s nimi. Výsledky jsou uvedeny pro slovník s 45000 přidanými slovními spojeními. Tabulka ukazuje snížení počtu chybně rozpoznaných krátkých slov.

6.4.5 Vyhodnocení

Experimentální výsledky potvrdily, že přidáním vhodných slovních spojení lze zvýšit úspěšnost rozpoznávání z 74.48 % na 77.94 %, i když jsou spojení přidávána plně automaticky. Je také patrná saturace v počtu přidávání slov, kdy více jak 45000 přidaných slov již nepřispívá ke zvýšení úspěšnosti rozpoznávače.

Výběr pomocí četnosti výskytu slovního spojení v textovém korpusu byl pro uvedenou úlohu vhodnější, neboť PMI nevybírá dostatečně četná slovní spojení. Vzájemná informace může pomoci při ručním výběru takových slovních spojení,

Tabulka 6.10: Nejčetnější chyby se slovníkem bez slovních spojení a se slovníkem se 45000 slovními spojeními.

Chyba	četnost výskytu		
	45000 slovních párů	žádné slovní páry	snížení počtu chyb
Inzerce "a"	44	59	15
Delece "a"	29	42	13
Delece "je"	27	34	7
Inzerce "v"	14	27	13
Delece "to"	23	22	-1
Inzerce "i"	15	21	6
Delece "v"	13	20	7
Delece "se"	16	17	1
Delece "na"	16	17	1
Inzerce "se"	5	14	9
Delece "z"	11	14	3
Delece "i"	7	11	4
Inzerce "na"	8	10	2
Inzerce "to"	6	9	3
Delece "s"	9	8	-1
Inzerce "z"	9	7	-2
Inzerce "je"	4	9	5
Delece "ho"	4	9	5
Delece "si"	5	8	3
Delece "co"	8	7	-1
Inzerce "o"	4	8	4
Inzerce "s"	2	8	6
Delece "by"	3	7	4

kdy je fonetická transkripce spojení běžnými fonologickými pravidly nesprávná, zejména u cizích slov.

Zvýšení úspěšnosti rozpoznávače přidáním slovních spojení má však za následek zvětšení slovníku a jazykového modelu, neboť jsou přidávána spojení tvořená nejčetnějšími slovy. Doba rozpoznávání je kvůli většímu jazykovému modelu delší.

Kapitola 7

Tvorba jazykového modelu

S rostoucí velikostí slovníku v systémech rozpoznávání řeči jsou si slova akusticky bližší, což je podpořeno i použitím akustických modelů, které se snaží vystihnout výslovnostní variabilitu mluvčích. Jazykový model obsahuje informace o struktuře jazyka, četnosti výskytu slov, sekvencích slov, či morfologických třídách. Jazykový model též definuje pravděpodobnost přechodu mezi slovy vyjádřenými skrytými markovskými modely.

Během rozpoznávání generuje rozpoznávač hypotézy (sekvence slov). Rozpoznaný text je pak hypotéza s nejvyšší pravděpodobností. Ohodnocování hypotéz je prováděno na základě akustického a jazykového modelu. Jazykový model pomáhá vybrat nejpravděpodobnější hypotézu příslušející nahranému zvukovému signálu mezi akusticky velmi podobnými hypotézami. Ohodnocování hypotézy akustickými modely je v současné době nejčastěji prováděno pomocí skrytých markovských modelů. Výsledkem je tedy jedna hodnota udávající pravděpodobnost hypotézy pro zachycený zvukový signál dané vztahem

$$P(akusticky\_model) = P(W|\mathbf{x}(t_1), \ldots, \mathbf{x}(t_n)), \tag{7.1}$$

kde W je sekvence slov převedená na fonémy tvořící jednu hypotézu a $\mathbf{x}(t_i)$ jsou vektory vytvořené parametrizací rozpoznávaného akustického signálu. Aby bylo možno jednoduše kombinovat akustickou a jazykovou část jazyka je nutné, aby jazykový model též ohodnotil hypotézu její pravděpodobností. Pak je celková pravděpodobnost hypotézy dána jednoduchým součinem obou pravděpodobností dle vztahu (7.2),

$$P(hypotezy) = P(akusticky\_model).P(jazykovy\_model) \tag{7.2}$$

neboť lze oba modely považovat za vzájemně nezávislé.

Aby byl výpočet pravděpodobnosti hypotézy co nejrychlejší a eliminovaly se problémy s aritmetikou malých čísel je místo pravděpodobnosti počítán její loga-

ritmus, pak

$$\ln(P(hypotezy)) = \ln(P(akusticky\_model)) + \ln(P(jazykovy\_model)). \tag{7.3}$$

Při hledání nejpravděpodobnější hypotézy je pravděpodobnost používána jako metrika. Nejlepší hypotéza však může být vybrána na základě libovolné ohodnocovací funkce F. Modifikací vztahu (7.3) dostaneme oblíbenou ohodnocovací funkci používanou například v rozpoznávačích vyvinutých v Laboratoři počítačového zpracování řeči na Technické univerzitě v Liberci, či v balíku HTK [4]. Ohodnocovací funkce je definována následovně

$$F(hypotezy) = \ln(P(akusticky\_model)) + \beta \ln(P(jazykovy\_model)), \tag{7.4}$$

kde β je váha jazykového modelu.

Implementace rozpoznávačů spojité řeči v reálných aplikacích musí pracovat co nejrychleji a s omezenými požadavky na paměť počítače. Proto je nutné aplikovat prořezávání hypotéz, kdy je během rozpoznávání uchováváno a dále prohledáváno jen omezené množství dosud nejpravděpodobnějších hypotéz. Z tohoto důvodu je nutné, aby bylo možné určit pravděpodobnost jen části hypotézy odpovídající doposud zpracovanému zvukovému signálu.

V současné době existují dva hlavní typy jazykových modelů. První model je založen na pravděpodobnostní či deterministické bezkontextové gramatice, viz [5]. Pro každé pravidlo pravděpodobnostní bezkontextové gramatiky je odhadnuta jeho pravděpodobnost z textového korpusu. Tento model je velice obtížné vytvořit, neboť v přirozeném jazyce existuje spousta výjimek obtížně popsatelných formální bezkontextovou gramatikou.

Druhý přístup je založen na četnostech sekvencí slov počítaných z korpusu, takzvaný n-gramový jazykový model. Jazykový model může být odvozen jak ze slov, tak i z dalších morfologických tříd. Pak je zjišťována četnost sekvencí těchto tříd [27]. Jazykový model odvozený z neslovních jednotek je pak většinou nutné pro použití v rozpoznávači přepočítat na slovní jazykový model.

V současné době je nejpoužívanějším jazykovým modelem pro rozpoznávání řeči slovní n-gramový jazykový model. Tento model je používán i v rozpoznávačích použitých v této práci. Model tvoří rozložení pravděpodobnosti výskytů sekvencí slov. Pravděpodobnost sekvence slov $\mathbf{W} = w_1, w_2, \ldots, w_s$ je pak dána vztahem

$$\begin{aligned} P(\mathbf{W}) &= P(w_1, w_2, \ldots, w_s) \\ &= P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)\ldots P(w_s|w_1, \ldots, w_{s-1}) \end{aligned} \tag{7.5}$$

Pravděpodobnost dle předchozího vztahu není možné ve skutečnosti odhadnout,

proto je $P(\mathbf{W})$ aproximována vztahem

$$P(\mathbf{W}) = \prod_{i=1}^{s} P(w_i| \underbrace{w_{i-1}, \ldots, w_{i-n}}_{historie\ delky\ n}), \tag{7.6}$$

kdy je pro každé slovo bráno v úvahu jen $n-1$ předchozích slov. Podmíněná pravděpodobnost $P(w_i|w_{i-1}, \ldots, w_{i-n})$ se nazývá n-gram. Odhad n-gramů je prováděn z velkého textového korpusu dle vztahu

$$P(w_i|w_{i-1}, \ldots, w_{i-n}) = \frac{C(w_i, w_{i-1}, \ldots, w_{i-n})}{C(w_{i-1}, \ldots, w_{i-n})}, \tag{7.7}$$

kde C je absolutní četnost výskytu příslušné sekvence slov. Pro výpočet modelu jsou brána jen slova ze slovníku.

V dnešní době je pro menší slovníky používán trigramový jazykový model (trojice slov) a pro větší slovníky 100000 a více slov je v rozpoznávačích používán bigramový jazykový model. Důvodem nižšího řádu jazykového modelu jsou vysoké výpočetní a paměťové nároky při vytváření jazykového modelu a při rozpoznávání.

Problémem při použití n-gramového modelu jsou neviděné sekvence slov. Taková sekvence má četnost 0, tedy příslušný n-gram má též nulovou hodnotu a všechny sekvence slov obsahující neviděnou sekvenci by byly nepravděpodobné. Tomuto jevu se lze vyhnout takzvaným vyhlazováním, kdy se nulové bigramy nahradí malým nenulovým číslem. Existuje několik metod vyhlazování, viz[5].

Často používaná metoda je Witten-Bell [29] daná vztahem

$$P(w_n|w_{n-1}) = \frac{c(w_{n-1}w_n)}{c(w_{n-1}) + N(w_{n-1})}\text{ , když } c(w_{n-1}w_n) > 0 \tag{7.8}$$

$$= \frac{N(w_{n-1}}{(V - N(w_{n-1}))(c(w_{n-1}) + N(w_{n-1}))}\text{ jinak }, \tag{7.9}$$

kde $c()$ jsou absolutní četnosti výskytu sekvencí slov v korpusu, $N(w_{n-1})$ je počet různých následníků slova w_{n-1} a V je počet slov ve slovníku.

Pokud $N(w_{n-1}) > \frac{V}{2}$, pak odhad bigramu dle vztahu (7.9) je větší než odhad dle vztahu (7.8), tedy neviděný bigram má vyšší hodnotu než viděný [52]. V tomto případě je v rozpoznávači vyvinutém v Laboratoři počítačového zpracování řeči použita jednodušší metoda přičtení 1 ke všem absolutním četnostem výskytu.

$$P(w_n|w_{n-1}) = \frac{c(w_{n-1}w_n) + 1}{c(w_{n-1}) + V}. \tag{7.10}$$

Pokud existuje slovo ve slovníku, které nemá žádné viděné sousedy, pak je hodnota jeho bigramů odhadnuta dle vztahu

$$P(w_n|w_{n-1}) = \frac{1}{V}. \tag{7.11}$$

Tento případ může nastat, když je do slovníku manuálně vloženo nějaké slovo, nebo je slovník vytvořen z jiných textů, než z kterých je počítán jazykový model.

Pro ilustraci vlivu jazykového modelu na úspěšnost rozpoznávání byl proveden experiment, kdy byl jazykový model z rozpoznávače spojité řeči odstraněn. Experiment byl proveden na databázi TV2005, sekce 4.2.2. Výsledky jsou uvedeny v tabulce 7.1.

Tabulka 7.1: Vliv jazykového modelu na úspěšnost rozpoznávání.

	úspěšnost rozpoznávání (Acc)
S jazykovým modelem	80.06 %
Bez jazykového modelu	40.47 %

7.1 Výpočet jazykového modelu

Pro slovník o velikosti 312000 slov, který je v současné době používán rozpoznávačem spojité řeči použitém v této práci, by velikost bigramového jazykového modelu mohla být až 312000^2 bigramů. Pokud by byly všechny bigramy viděné a každý bigram by zabíral 32 bitů paměti (pole 312000×312000 integerů), pak by celý jazykový model zabíral 362 GB. Takové množství paměti není dnes běžně dostupné. Přestože nebudou některé bigramy v textovém korpusu viděny, lze předpokládat, že množství různých viděných bigramů bude značné díky dosti volnému pořadí slov v české větě. To je významný rozdíl oproti jazykům s pevnou stavbou věty jako je angličtina nebo němčina.

Pro výpočet jazykového modelu již existuje několik nástrojů. Nejznámější je SRILM toolkit [30]. Tento software je univerzální produkt pro výpočet mnoha druhů jazykových modelů. Řád jazykového modelu je omezen pouze velikostí instalované paměti. SRILM nabízí mnoho druhů vyhlazování jazykového modelu. SRILM je schopen spočítat jazykový model pro slovník používaný pro rozpoznávání češtiny. V Laboratoři počítačového zpracování řeči není používán z důvodu nevhodné licence pro použití v komerčních produktech.

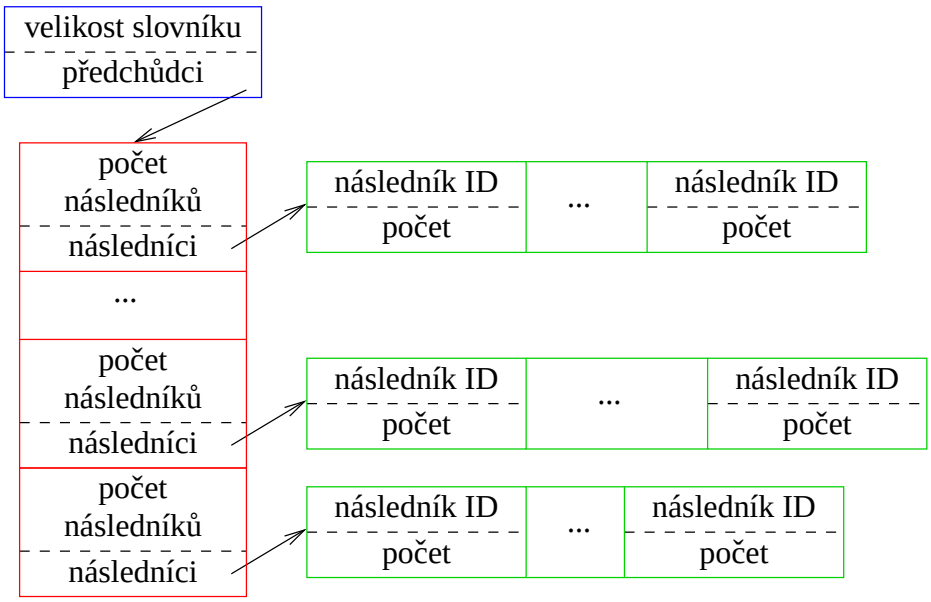
7.1.1 Implementace výpočtu bigramů

Z předchozího textu je zřejmé, že jazykový model bude možné přijatelně rychle vytvořit pouze tehdy, když bude viděných bigramů tolik, kolik se vejde do paměti. Možnost ukládat mezivýsledky na pevný disk počítače je nevyhovující, neboť tak dochází k nepřijatelnému časovému prodloužení výpočtu.

Implementace byla provedena v jazyce C, který umožňuje efektivně alokovat paměť. Byly vyzkoušeny 2 přístupy. V prvním přístupu byla primárním požadavkem minimální paměťová náročnost. V druhém přístupu byla upřednostňována rychlost výpočtu. Vždy je nutné zaznamenat a uchovávat všechny viděné dvojice slov. V obou případech je slovník načten do pole řetězců znaků a dále je již počítáno jen s indexy slov v tomto poli.

Lineární struktura

Datová struktura pro uchovávání dvojic slov s minimálními paměťovými nároky je uvedena na obrázku 7.1.



Obrázek 7.1: Datová struktura pro uchovávání dvojic slov s minimálními paměťovými nároky.

Všichni následníci stejného slova jsou se svými četnostmi výskytu za tímto předchůdcem uchováváni v poli následníků. Velikost každého následníka je 8 bytů (4 byty index následníka, 4 byty četnost). Počet předchůdců je stejný jako počet slov ve slovníku. Velikost předchůdce je pro 32bitové procesory taktéž 8 bytů (4 byty počet následníků, 32 bitů adresa pole s následníky).

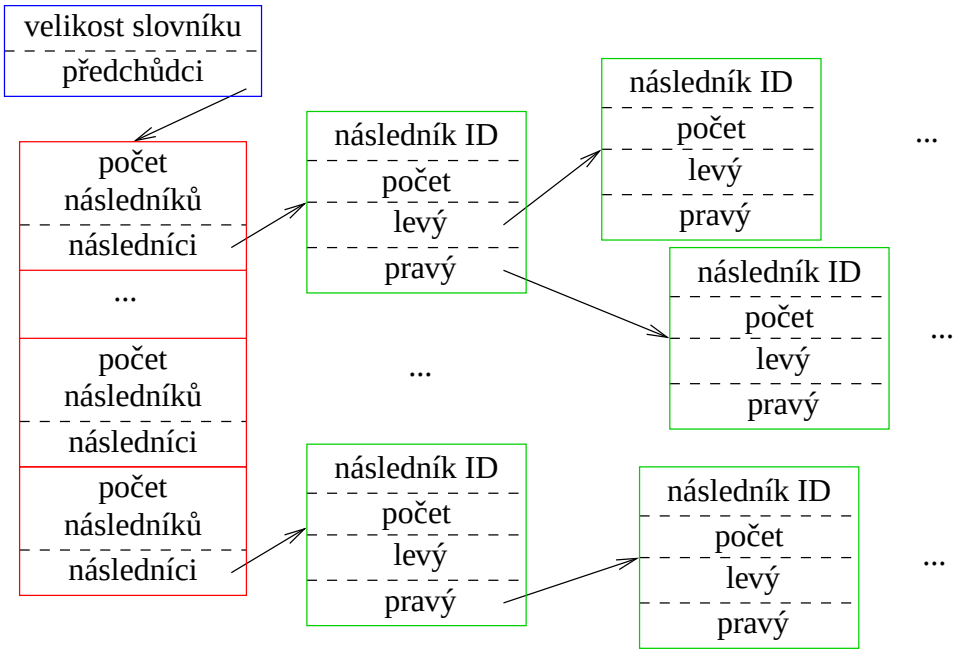
Tato struktura je méně výhodná pro přidávání nového následníka, neboť jsou následníci uloženi v souvislém úseku paměti. Při přidávání je nutné alokovat nový souvislý kus paměti, aby se do něj vešlo původní pole následníků a nový následník. Alokace mnoha velkých souvislých úseků paměti je pomalá, neboť je již většinou fragmentována jinými programy a souvislý kus paměti je nutno delší dobu

hledat.

Úprava četnosti výskytu následníků se v této struktuře provádí sekvenčním prohledáváním a následnou úpravou četnosti nalezeného následníka. Pro zvýšení rychlosti modifikace četností výskytu následníků je dobré, aby četnější následníci byly na začátku seznamu, což lze docílit občasným setříděním. Binární dělení je další možná alternativa přístupu k setříděným následníkům.

Stromová struktura

Datová struktura pro uchovávání dvojic slov s požadavkem na maximální rychlost výpočtu je uvedena na obrázku 7.2.

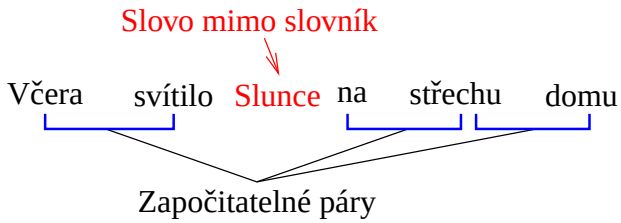


Obrázek 7.2: Datová struktura pro uchovávání dvojic slov s požadavkem na maximální rychlost výpočtu.

Oproti předchozí datové struktuře z obrázku 7.1 jsou následníci uloženi v binárním stromu, což umožňuje rychlejší vyhledávání než sekvenční přístup. Stejně jako v předchozím případě je dobré, aby četní následníci byli co nejblíže kořeni stromu. Velikost následníka je pro 32bitové procesory o 8 bytů větší díky dvěma adresám ukazujícím na levý a pravý podstrom. Paměťová alokace je jednodušší a rychlejší, neboť není nutné alokovat velké souvislé bloky paměti při přidávání nového následníka.

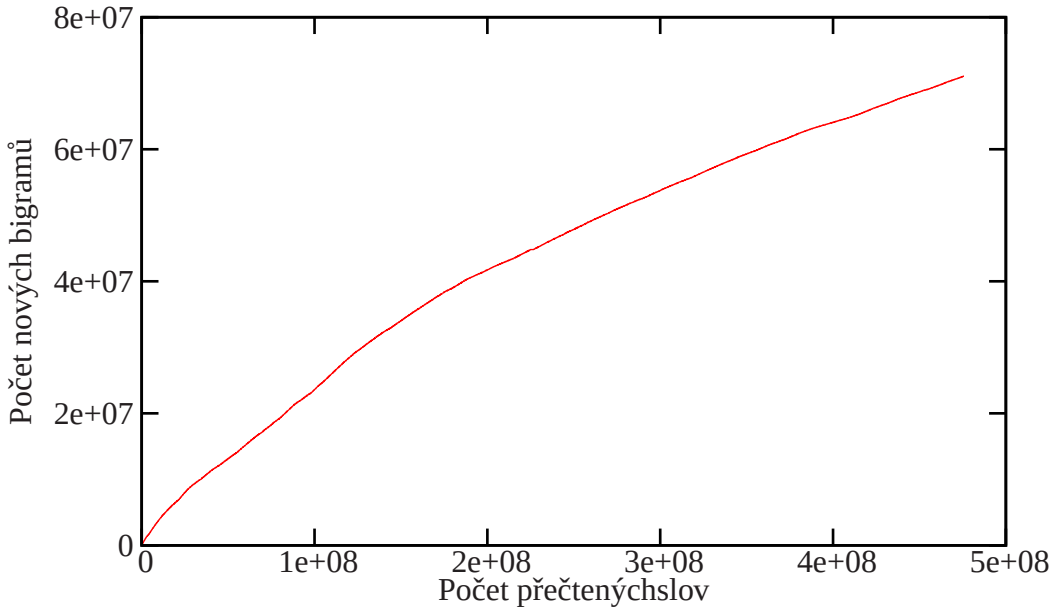
7.1.2 Experimenty

Pro porovnání uvedených datových struktur při výpočtu bigramového modelu byl proveden výpočet jazykového modelu se slovníkem o velikost 312 tisíc slov z korpusu uvedeného v kapitole 5. Korpus obsahuje 3.5 GB textu. Během výpočtu bylo viděno 458362779 započitatelných slovních dvojic, z nichž 71083693 bylo různých, což je 0.07% z možných 312000^2. Příklad započitatelných slovních dvojic je uveden na obrázku 7.3.



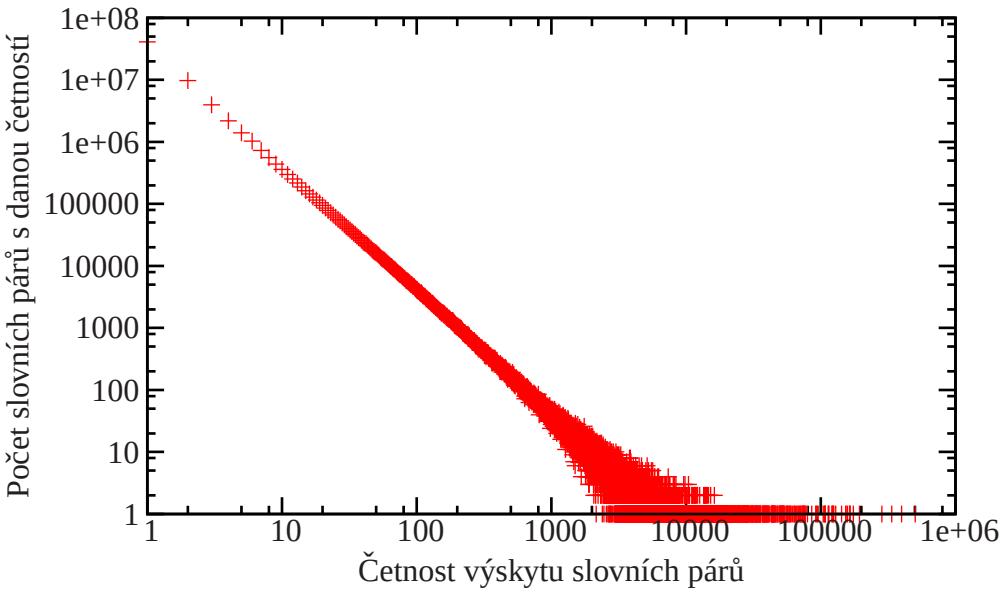
Obrázek 7.3: Příklad započitatelných slovních párů.

Průběh výpočtu bigramového jazykového modelu z textového korpusu je uveden na obrázku 7.4. Ze stále stoupající tendence nalézání nových slovních párů



Obrázek 7.4: Průběh výpočtu jazykového bigramového jazykového modelu

je patrné, že i tak velký korpus neobsahuje většinu započitatelných slovních párů, které se pro daný slovník vyskytují. Je patrné pouze mírné snižování rostoucí tendence počtu neviděných započitatelných slovních párů.



Obrázek 7.5: Histogram četností výskytu slovních párů v textovém korpusu. Graf je vykreslen v logaritmických souřadnicích.

Obrázek 7.5 zobrazuje histogram počtu četností viděných párů. Je patrné, že uvedené rozložení je podobné rozložení četností samotných slov. Tudíž různých párů s nízkou četností je mnoho a párů s velikou četností je málo. Průběh výpočtu jazykového modelu na obrázku 7.4 indikuje nepřesný odhad bigramů pro málo četné slovní páry v důsledku nedostatku dat. Řešením je sběr dalších dat, kterých je pro lepší odhad málo četných bigramů, potřeba veliké množství.

Čas výpočtu modelu a spotřebovaná paměť pro různé datové struktury reprezentující slovní dvojice jsou uvedeny v tabulce 7.2. Je vidět, že použití stromové struktury vede k rychlejšímu výpočtu, přičemž paměťová náročnost nepřekračuje možnosti běžných osobních počítačů, ve kterých lze využít až 3.3 GB[1] operační paměti. Experimenty byly prováděny na konfiguraci: Intel Pentium 4 HT 3 GHz, 3 GB RAM.

Implementace se stromovou strukturou spotřebuje více než dvojnásobek paměti vyžadovaný lineární strukturou. Procházení stromové struktury je implementováno pomocí zásobníku, který spotřebuje nezanedbatelnou část paměti. Použití rekurze nebylo možné díky nedostatečně velkému zásobníku, na který se skládají návratové adresy. SRILM toolkit je univerzální nástroj pro výpočet jazykového modelu, proto v něm nemohou být použity optimalizace vhodné pouze pro výpo-

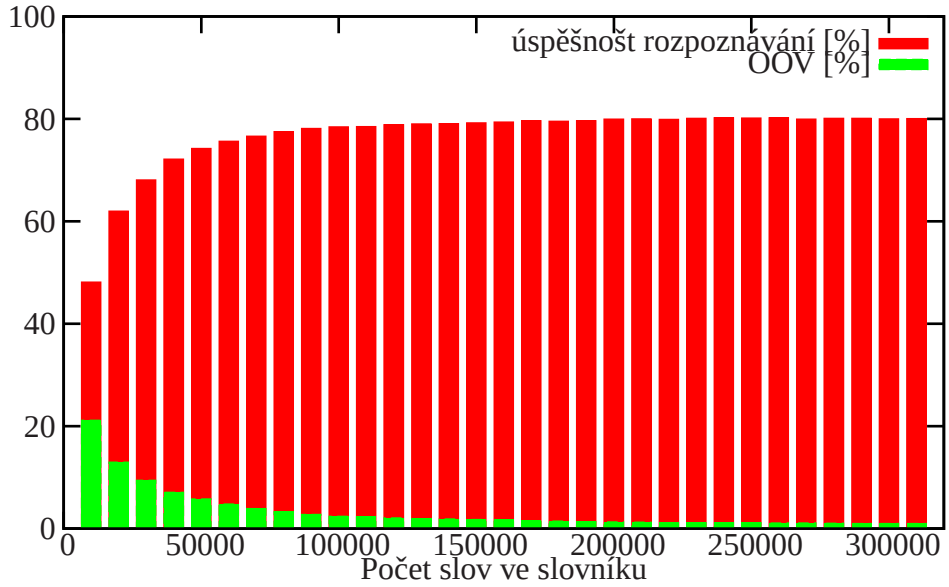
[1]Běžné osobní počítače lze osadit 4 GB operační paměti. Většina chipsetů ale rezervuje část adresovatelného prostoru 4 GB pro potřeby připojených zařízení: grafická karta, můstky, přídavné karty, ...

Tabulka 7.2: Vliv datové struktury bigramového modelu na rychlost výpočtu a spotřebovanou paměť.

Jazykový model	lineární struktura	stromová struktura	SRILM
Čas výpočtu [hod:min:sek]	2:50:45	0:37:15	1:15:01
Spotřebovaná paměť	601 MB	1639 MB	1780 MB

čet bigramového modelu. Výpočet je proto delší než při použití stromové struktury a spotřebuje více paměti než stromová struktura. Přesto je však SRILM pro češtinu použitelný.

Další experiment ukazuje vliv velikosti slovníku na úspěšnost rozpoznávání. Pro různě velké slovníky je v grafu 7.6 uvedena úspěšnost rozpoznávání a počet slov mimo slovník (OOV). Experiment je proveden na databázi TV2005 4.2.2.



Obrázek 7.6: Vliv velikosti slovníku na úspěšnost rozpoznávání.

Další experiment provedený na téže databázi TV2005 4.2.2 ukazuje vliv interpunkce na počítání jazykového modelu. Obrázek 7.3 ukazuje, jak může být započtení slovního páru přerušeno slovem, které není ve slovníku. Podobná situace může nastat, kdy jsou dvě slova oddělena interpunkcí. Jsou vypočítány dva jazykové modely, přičemž v jednom interpunkce znemožní započítání slovního páru, který je interpunkcí přerušen. Při počítání druhého jazykového modelu je interpunkce ve větě ignorována a všechny slovní páry ve větě jsou započítány. Tabulka 7.3 ukazuje úspěšnost rozpoznávání pro oba případy.

Tabulka 7.3: Vliv interpunkce při výpočtu jazykového modelu na úspěšnost rozpoznávání.

	úspěšnost rozpoznávání (Acc)
Interpunkce přerušuje slovní pár	78.04 %
Interpunkce je ignorována	80.06 %

Z výsledků je vidět, že pokud interpunkce znemožňuje započítání slovního páru do jazykového modelu, je výsledný jazykový model horší. Věta je většinou vyslovena jako celek bez ohledu na interpunkci, což se projevilo i v uvedeném experimentu.

7.2 Zhodnocení

V této kapitole jsou navrženy a implementovány potupy pro výpočet bigramového jazykového modelu. Byla implementována varianta s důrazem na minimální spotřebu paměti a varianta s maximální rychlostí výpočtu. Obě varianty spočítají jazykový model v přijatelném čase a se splnitelnými paměťovými nároky. Velikost výsledného jazykového modelu a spotřebovanou paměť počítače lze pro nový veliký slovník odhadnout jen přibližně. Proto je v případě nedostatku paměti nutné použít méně paměťově náročnou variantu, která je však mnohem rychlejší než swapování, které by nastalo při nedostatku paměti. Pokud dojde při výpočtu ke swapování, může se výpočet protáhnout i na několik dní. Uvedené experimenty také ukazují rostoucí tendenci nalézání nových slovních párů pro slovník o velikosti 312 tisíc slov i v korpusu, který obsahuje 3.5 GB textu, což se projeví v nižší úspěšnosti rozpoznávání málo četných slovních párů, neboť jsou jejich n-gramy odhadnuty méně přesně.

Přestože v experimentu ukazujícím vliv velikosti slovníku na úspěšnost rozpoznávání od slovníku obsahujícího 200000 slov již úspěšnost výrazně neroste, lze očekávat, že na jiných datech by úspěšnost mohla být vyšší pro větší slovníky. Testovací data jsou televizní zprávy, které obsahují jen omezený počet slov vztahující se k danému tématu, což ukazuje i počet slov mimo slovník. Pokud má být rozpoznáno slovo, které není ve slovníku, pak rozpoznávač nemá šanci slovo rozpoznat. Větší slovníky dávají rozpoznávači šanci rozpoznat více slov.

Kapitola 8

Analýza výstupu rozpoznávacího systému

Výsledky rozpoznávání řeči jsou nejčastěji vyjádřeny úspěšností rozpoznávání, nebo mírou chybovosti, jak je uvedeno v sekci 4.3. Vyhodnocování rozpoznávání spojité řeči je oproti rozpoznávání izolovaných slov složitější v tom, že kromě špatně rozpoznaných slov (substituce s) mohou být některá slova rozpoznávačem ignorována (delece d) a jiná přidána (inzerce i) oproti referenčnímu textu. Porovnávání referenčního textu a rozpoznané řeči se provádí zarovnáváním, které je založeno na dynamickém programování využívajícím Bellmanova principu optimality. Hledá se tedy cesta mapující rozpoznanou řeč na referenční text za minimální cenu. Správně rozpoznané slovo (hit h) nezvyšuje cenu cesty, substituce zvýší cenu cesty o 10 a delece nebo inzerce o 7. Je možné volit i jiné ceny substituce, delece a inzerce, ale aby nebyla substituce nahrazována sekvencí delece a inzerce, musí platit, že

$$c(s) < c(d) + c(i), \tag{8.1}$$

kde $c()$ označuje cenu.

Vyhodnocování výsledků rozpoznávání není stále úplně vyřešeno. Je možné s jistou úspěšností zjistit jaká slova byla rozpoznána správně, nahrazena, přidána nebo ignorována. Úspěšnost této klasifikace chyb je závislá na úspěšnosti rozpoznávání samotném. Pokud je promluva rozpoznána s malým počtem chyb, podaří se přesně identifikovat jednotlivé typy chyb. Pokud je ale promluva rozpoznána s vysokým počtem chyb, například v důsledku nízké kvality signálu, je přiřazení konkrétních chyb jednotlivým slovům diskutabilní. Je pak pouze zřejmé, kolik chyb se ve výsledku rozpoznávače vyskytuje.

Dalším dosud nevyřešeným problémem je významnost chyb, kdy není rozdíl, jestli rozpoznávač udělá chybu, která neovlivní informaci obsaženou ve větě, nebo je informace ve větě díky chybě rozpoznávače poškozena. Tato sémantická vrstva

jazyka není zatím do rozpoznávačů mluvené řeči zakomponována.

V následujících ukázkách jsou uvedeny skutečné chyby nalezené ve výstupu rozpoznávače. Pokud dojde k záměně jediného slova, pak je toto slovo akusticky velmi podobné, například:

Reference:	Jiří Paroubek **odmítl** návrhy ODS komentovat.
Rozpoznáno:	Jiří Paroubek **odmítnul** návrhy ODS komentovat.

Akusticky podobné slovo může však být významově rozdílné.

Reference:	Podle **našeho** názoru by se měli obrátit na policii.
Rozpoznáno:	Podle **vašeho** názoru by se měli obrátit na policii.

Občas odlišný význam chybného slova nemusí vadit.

Reference:	Je **tu** skvělá atmosféra **miluji** tyhle show.
Rozpoznáno:	Je **to** skvělá atmosféra **miluju** tyhle show.

V některých případech dochází vlivem nesprávného tvaru slova k pozastavení čtenáře. V tomto případě zpětně zjišťuje, kdo se *dovolává*.

Reference:	Určitě bude těžší se **dovolávat** spravedlnosti.
Rozpoznáno:	Určitě bude těžší se **dovolává** spravedlnosti.

Nejčetnější chyby jsou způsobeny vynecháním nebo přidáním krátkých slov jako jsou předložky a spojky.

Reference:	Místní jsou však proti. **I** proto se na dnešním veřejném …
Rozpoznáno:	Místní jsou však proti proto se na dnešním veřejném …

Jiný příklad vložení krátkého slova.

Reference:	že není zabezpečený náklad.
Rozpoznáno:	**to** že není zabezpečený náklad.

Vložení nebo vynechání i krátkého slova je nebezpečné, neboť se Viterbiho algoritmus snaží doplnit mezeru, či zkrátit slovo sousedící s vloženým slovem. Dochází tak k chybám typu *delece, substituce* či *inzerce, substituce*.

Reference:	Těšíme se **nashledanou**.
Rozpoznáno:	Těšíme se **na stranu**.

V předchozím případě je zároveň vidět, že chyba se může dostat i do referenčních přepisů. Správně má být na shledanou. Jiný případ chyby delece–substituce:

Reference:	**My jsme** v odpoledních hodinách zadrželi celkem …
Rozpoznáno:	**Slezsko** v odpoledních hodinách zadrželi celkem …

Přepis zarušených signálů, například hlukem na pozadí, je značně složitý a může vést k opakovaným chybám.

Reference:	Jé vy jste z Čech. V Praze jsem byl již před dlouhou dobou, Připomínala mi právě Londýn šedesátých let.
Rozpoznáno:	Jé vy jste se v Praze zeď jejíž předlohu do boky připomínala mi právě rodí z šedesátých letech.

V důsledku značného zarušení začátku signálu signálu byla při přepisu parlamentních promluv nalezena následující chyba:

Reference:	Pane předsedo, místopředsedo, vážené paní poslankyně, ...
Rozpoznáno:	Ne příliš vlastní velmi sexy dcera paní poslankyně, ...

Tato chyba rozpoznávače se může v rukou bulvárního tisku stát nebezpečným důkazem o sexuálním harašení na půdě poslanecké sněmovny.

8.1 Zarovnávání textů

Na obrázku 8.1 je uveden postup zarovnávání referenční věty „Na Internetu se objevila nahrávka s údajným hlasem.“ a rozpoznané věty „Na Internetu objevili přihrávku údajným hlasem.“ Horizontální a vertikální šipky označují deleci a inzerci. Diagonální šipky označují substituci nebo hit, pokud se slova v příslušném řádku a sloupci shodují. Fialová čára ukazuje nejlevnější cesty, jejichž cena je 34. Tabulka má rozměr větší, než je počet slov v referenci a rozpoznané promluvě, neboť výsledkem jsou způsoby přechodů mezi buňkami, kterých by bylo v menší tabulce o 1 méně, než je potřeba.

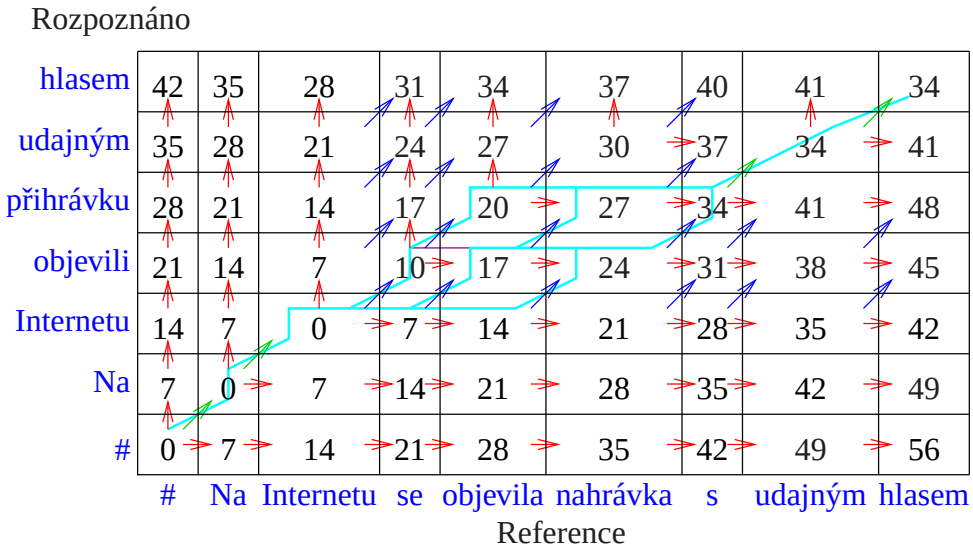
Nejlevnější cesty jsou: $hhddsshh$, $hhssddhh$, $hhdsdshh$, $hhsdsdhh$, $hhdssdhh$, $hhsddshh$. Je vidět, že pokud jsou vedle sebe inzerce a substituce či delece a substituce, pak na pořadí nezáleží, jsou rovnocenné.

Inzerce a delece se vedle sebe vyskytovat nemohou, což je zajištěno požadavkem (8.1). Zároveň je z požadavku (8.1) patrné, že zarovnávání produkuje pouze sekvence inzercí a substitucí nebo delecí a substitucí. Sekvence inzercí, substitucí a delecí se nemůže vyskytnout. Počet cest se shodnou cenou C je dán vztahem

$$C = \prod_{i=1}^{n} \frac{k_i!}{k_{si}!.k_{di}!}, \tag{8.2}$$

$$k_i = k_{si} + k_{di}, \tag{8.3}$$

kde k je délka i-té sekvence substitucí a delecí nebo substitucí a inzercí, k_{si} je počet substitucí v i-té sekvenci a k_{di} je počet delecí či inzercí v i-té sekvenci.



Obrázek 8.1: Zarovnávání textů pomocí dynamického programování.

Písmeno n je počet sekvencí substitucí a delecí případně substitucí a inzercí v zarovnávaném textu. Pořadí ostatních členů cesty je jednoznačné.

Úspěšnost rozpoznávání v příkladu je dle vztahu (4.2)

$$Acc = \frac{8 - 0 - 2 - 2}{8} = 50\%.$$

8.2 Detailní analýza

Situace, kdy se vedle sebe vyskytují právě substituce a inzerce nebo delece, nejsou ojedinělé, což je dáno způsobem rozpoznávání. Pokud během rozpoznávání dojde k inzerci či deleci, snaží se rozpoznávač doplnit akusticky nejvěrohodnější slovo, které je zároveň tak dlouhé, aby délkově co nejdříve eliminovalo vliv předcházející inzerce či delece. Totéž platí i obráceně, kdy rozpoznávač rozpozná slovo delší nebo kratší (jiná přípona či předpona), než ve skutečnosti má být. Pak si rozpoznávač musí pomoci následnou delecí či inzercí.

Často je dobré vědět, která slova jsou nejčastěji špatně rozpoznána, aby mohla být cíleně a efektivně zvyšována úspěšnost rozpoznávání. Chyba může být totiž způsobena nesprávným fonetickým přepisem ve slovníku. Úspěšnost rozpoznávání krátkých slov lze zvýšit vhodnými slovními spojeními, viz sekce 6.4.

Manuální kontrola rozpoznané řeči je v detailní analýze výsledků rozpoznávače nezbytná. Procházení výsledků je však velmi časově i fyzicky náročné. Pomoci může zarovnávání, pomocí kterého lze sestavit seznam nejčastějších chyb

tak, že je procházena vygenerovaná sekvence substitucí, inzercí, delecí a hitů zároveň s referenčním a rozpoznaným textem. Pokud je nalezena inzerce, je slovo vybráno z rozpoznané promluvy. Pokud je nalezena delece, je slovo vybráno z referenčního textu. V případě substituce a inzerce je slovo vybráno z obou textů a pokud není shodné v obou textech je prohlášeno za substituci, jinak za hit.

Více cest se stejnou cenou je v tomto případě nežádoucí, neboť bychom chtěli vědět, které slovo bylo substituováno a které vloženo, či eliminováno. Řešením je upravit vztah (8.1) tak, aby byla stále splněna nerovnost a zároveň, aby kratší slovo bylo spíše delece nebo inzerce a delší slovo bylo spíše substituce. Úprava cen inzerce, delece a substituce je provedena dle následujících vztahů

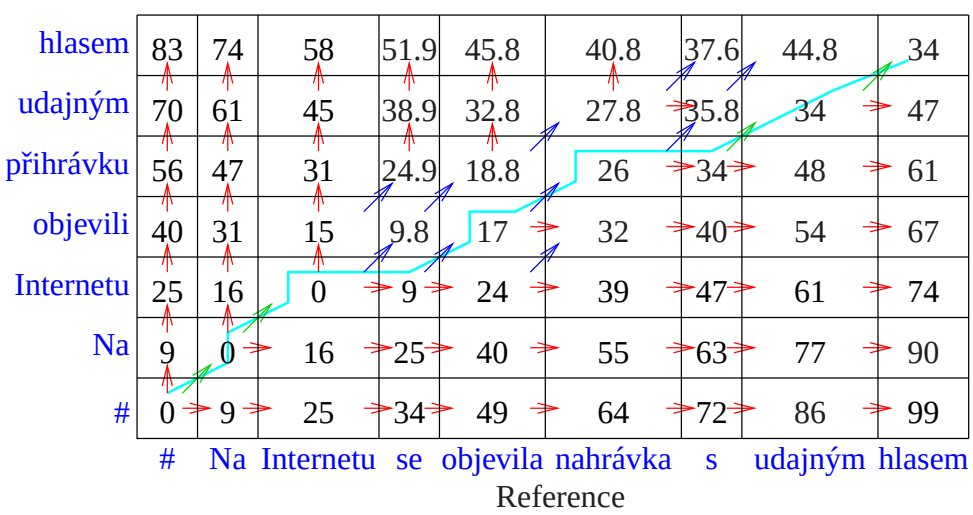
$$c_n(i) = c(i) + l(i), \tag{8.4}$$

$$c_n(d) = c(d) + l(d), \tag{8.5}$$

$$\begin{aligned} c_n(s) &= c(s) - \frac{1}{l_d}, \text{ když } l_d > 0, \\ &= c(s) - 2 \text{ jinak}, \end{aligned} \tag{8.6}$$

kde $c()$ a $c_n()$ jsou původní, respektive nové ceny přechodů, $l()$ je délka slova představující inzerce nebo deleci a l_d rozdíl délek substituovaných slov, $c(i) = c(d) = 7$, $c(s) = 10$.

Upravená tabulka z předchozího příkladu uvedeném v sekci 8.1 je ukázána na obrázku 8.2.



Obrázek 8.2: Zarovnávání textů pomocí dynamického programování s eliminací více cest.

Vhodně zvolené ceny přechodů eliminovaly všechny cesty až na jednu. Vý-

sledek je tedy jednoznačný $hhdssdhh$, což odpovídá představě o nejlepším zarovnání delecí a substitucí v tomto příkladu.

Případy, kdy bychom chtěli, aby zarovnání dalo sekvenci dsi, například pro referenční text „včera si dali vědět“ a rozpoznanou promluvu „včera dal s vědět“, uvedená změna cen přechodů neřeší. Výsledek zarovnávání bude ss. Aby bylo možno řešit takovéto případy, je nutné velmi opatrně porušit nerovnost (8.1).

8.3 Nejčetnější chyby rozpoznávání

Tato sekce uvádí nejčetnější chyby rozpoznávání spojitého rozpoznávače řeči se slovníkem 312 tisíc slov na databázi TV2005 popsané v sekci 4.2.2. Rozpoznané promluvy obsahují 275 inzercí, 357 delecí, 1316 substitucí a 9769 slov v referenčním textu, což dává úspěšnost rozpoznávání 80.06%.

Tabulka 8.1 ukazuje nejčetnější chyby spojitého rozpoznávače řeči. Je patrné, že nejvíce chyb je způsobeno krátkými slovy, které vytvářejí inzerce nebo delece. Důvodem je jejich podobnost s různými šumy. Při špatně rozpoznaném sousedním slově jsou šumy „vhodnými“ kandidáty na rychlé doplnění délky špatně rozpoznaného slova. Dalším důvodem chyby může být jazykový model, který má například vyšší pravděpodobnost výskytu spojení „byl a“ než „byla“ s rozpoznanými sousedními slovy.

Celkový počet delecí a inzercí slov, která jsou maximálně 2 znaky dlouhá, je 431, což je 68 % všech inzercí a delecí. Kdyby se takto krátké inzerce a delece nevyskytovaly, pak by úspěšnost rozpoznávání stoupla na 82.12%.

V češtině není odlišena výslovnost písmen *i* a *y*. Akusticky je psaní *i* a *y* naznačeno pouze v ryze českých slovech u slabik di, ti, ni. Nejčetnější chyby způsobené nesprávným *i* nebo *y* jsou uvedeny v tabulce 8.2. V tabulce jsou všechna *i* a *y* nahrazena *i*.

Nesprávně rozpoznaných *i* a *y* je 20. Takováto chybovost je způsobena převážně jazykovým modelem. Bez těchto substitucí by úspěšnost rozpoznávání byla 80.3 %.

Následující tabulka 8.3 ukazuje nejčetnější špatně rozpoznaná příčestí minulá ($\sim$l, $\sim$li, $\sim$la, ...). Písmeno následující za *l* v příčestí je v tabulce odstraněno, aby mohla být různá příčestí brána jako ekvivalentní.

Chybovost v příčestí je větší než chybovost v *i* a *y*. Celkový počet chyb v příčestí byl 68, zde již na chybovost má vliv jak jazykový tak i akustický model. Úspěšnost rozpoznávání bez chyb v příčestí by byla 80.8 %.

Tabulka 8.1: Nejčetnější chyby spojitého rozpoznávače řeči.

Četnost výskytu	chyba
56	inzerce „a“
31	delece „a“
31	delece „je“
30	delece „v“
22	delece „to“
21	delece „i“
14	inzerce „v“
14	inzerce „z“
13	inzerce „i“
12	delece „na“
12	inzerce „je“
9	delece „že“
9	inzerce „o“
7	delece „do“
7	inzerce „k“
6	delece „už“
6	delece „z“
6	inzerce „za“

Tabulka 8.2: Substituce způsobené y/i.

Četnost výskytu	chyba
1	demonstrovali
1	skončili
1	museli
1	milí
1	dostali
1	ti
1	vyhrožovali
1	jezdili
1	pokusili
1	čerství
1	ovládli
1	přišli
1	ozvali
1	mohli

Tabulka 8.3: Nejčetnější chyby v příčestí minulém.

Četnost výskytu	chyba
4	dostal
2	otevřel
2	mohl
2	půjčil
2	byl
1	zkomplikoval
1	demonstroval
1	vihrožoval
1	potřeboval
1	nestranil
1	půjčoval
1	ukončil
1	pokusil

8.4 Zhodnocení

Nově navržená metoda detailní analýzy výsledků lépe přiřadí inzerce, delece a substituce ke konkrétním slovům, čímž je umožněno cílené zlepšování rozpoznávače. Metoda však stále nemusí přiřadit vždy takové typy chyb, jaké bychom očekávali, například sekvenci delece–substituce–inzerce. Proto jsou počty chyb uvedených v tabulkách přibližné počty skutečných chyb. Detailnější manuální kontrola výsledků zarovnávání ukázala několik případů substituce spojky *a* za mnohem delší slova. Takové případy byly však ojedinělé.

Výsledky nové metody také potvrzují předchozí domněnku, že značné množství chyb je způsobeno krátkými inzercemi a delecemi.

Kapitola 9

Adaptace jazykového modelu

Správný jazykový model má reflektovat jazyk, kterým se mluví a který je následně rozpoznáván. Pokud chceme rozpoznávat tématické promluvy, jako je například jednání parlamentu, lékařské zprávy, sportovní přenosy, atd., je nutné vytvořit nový jazykový model nebo upravit existující. Problém tématických promluv je malé množství dostupných dat, ze kterého by bylo možné spolehlivě odhadnout bigramy jazykového modelu. Adaptace jazykového modelu se snaží s malým množstvím nových dat upravit existující model tak, aby odpovídal novým požadavkům. Kromě tématických promluv je dalším důvodem adaptace časová změna jazykového modelu v televizních a rozhlasových zprávách.

Požadavek na nejlepší adaptovaný model lze také vyjádřit tak, že perplexita nového jazykového modelu na testovacích promluvách má být minimální.

Perplexita jazykového modelu na datech T je dána vztahem

$$PP(T) = 2^{H(T)}, \tag{9.1}$$

kde H(T) je entropie jazykového modelu. Entropie bigramového jazykového modelu na datech T je dána vztahem

$$H(T) = -\sum_{gr(T)} P(w_n|w_{n-1}) \log(P(w_n|w_{n-1})), \tag{9.2}$$

kde $gr(T)$ jsou všechny bigramy v textu T.

Požadavek na minimální perplexitu lze také chápat tak, že nový model by měl generovat data T s maximální pravděpodobností.

Přestože se daří výrazně snižovat perplexitu nových modelů, k znatelnému zlepšení úspěšnosti rozpoznávání s novými jazykovými modely dochází zřídka [32, 33, 34, 35]. Čím více je nový adaptovaný jazykový model odlišný od původního tím je adaptací dosahováno vyšší zlepšení.

Adaptace slovníku spočívá v přidání častých slov, která se objevují v novém korpusu, případně odebrání slov, která se objevují velmi zřídka, aby došlo ke zrychlení rozpoznávání.

Tato kapitola neuvádí žádné nové metody adaptace, ale zkoumá vybrané metody a porovnává vliv adaptace jazykového modelu a slovníku v různých případech. Testy mnoha různých jazykových modelů jsou umožněny výrazným zvýšením rychlosti vytváření jazykových modelů, viz kapitola 7.1.1 a použitím distribuované verze rozpoznávače vyvinutého na technické univerzitě v Liberci.

9.1 Metody adaptace jazykového modelu

V literatuře se objevuje více metod adaptace jazykového modelu. Od nejjednodušší lineární interpolace [36], log-lineární interpolace [37], maximum a posteriori (MAP) adaptace vycházející z metod adaptace používané na akustické modely [38], adaptace založené na principu maxima entropie [39], po různé ad-hoc metody. V následujícím jsou podrobněji popsány první dvě metody.

Nejběžnější a nejjednodušší metoda adaptace je lineární interpolace daná vztahem

$$\begin{aligned} P(w|h) &= \sum_{i=1}^{n} \lambda_i P_i(w|h), \\ \sum_{i=1}^{n} \lambda_i &= 1 \end{aligned} \tag{9.3}$$

kde $P_i(w|h)$ jsou bigramy jazykových modelů, ze kterých je nový model adaptován a λ_i je váha, která je odhadována z held-out dat tak, aby perplexita nového modelu byla na těchto datech minimální. Odhad λ_i může být proveden všeobecně známým EM algoritmem.

Log-lineární interpolace je dána vztahem

$$P(w|h) \equiv \prod_{i=1}^{n} P_i(w|h)^{\lambda_i}. \tag{9.4}$$

Oproti lineární interpolaci nemusí být výsledný bigram v intervalu $\langle 0, 1 \rangle$, což rozpoznávačům většinou nevadí. Log-lineární interpolace zvýší hodnoty četných bigramů v novém modelu více a sníží hodnotu méně četných bigramů více než lineární interpolace.

Obdobného efektu jako při interpolaci n-gramů lze dosáhnout přímo interpolací absolutních četností výskytu n-tic slov.

Jazykové modely používané k adaptaci jsou vytvořeny z malých korpusů. Dokumenty tvořící malé korpusy se v případě tématické týkají jednoho tématu a

jsou vybrány manuálně, nebo automaticky. Automatické zařazování dokumentů může být provedeno na základě vzdálenosti od manuálně vybraných vzorků pomocí četnosti výskytu slov v dokumentu nebo míry TFIDF běžně používané pro klasifikaci dokumentů [5]. Plně automatické rozdělování dokumentů je prováděno některou z metod shlukování. Výsledný model je nejčastěji vytvořen lineární interpolací tématických jazykových modelů a všeobecného jazykového modelu vytvořeného z velkého množství například novinových textů. Všeobecný model se přidává právě proto, že díky velkému množství textů, ze kterých byl vytvořen, jsou n-gramy běžného jazyka odhadnuty spolehlivěji.

9.2 Časová adaptace jazykového modelu systému rozpoznávání zpráv

Přepis televizních a rozhlasových zpráv je v poslední době rychle se vyvíjející část počítačového zpracování řeči. Je k dispozici poměrně kvalitní akustický signál ze studií. Jazykový model zpráv je podobný jazykovému modelu zpráv v novinách, které jsou snadno dostupné na internetu a jejich získávání lze automatizovat, viz sekce 5.1. O přepisy zpráv je také zájem v komerční oblasti.

Témata zpráv se v průběhu času mění, proto je vhodné jazykový model neustále doplňovat o nové texty z novin, či přímo přepisy starších zpráv. Tato sekce ukazuje vliv přidávání nových textů na úspěšnost rozpoznávání během zvoleného časového úseku.

Metody časové adaptace jazykového modelu jsou v podstatě shodné s tématickou adaptací, jen tématické korpusy jsou nahrazeny korpusy z různých časových období a časově vzdálenější korpusy mají nižší váhu.

9.2.1 Experimenty a zhodnocení

Experimenty jsou prováděny na systému pro rozpoznávání televizních a rozhlasových zpráv vyvíjeném v Laboratoři počítačového zpracování řeči Technické univerzity v Liberci [2]. Tento systém obsahuje rozpoznávač spojité řeči a pracuje se slovníkem obsahujícím 312 tisíc slov a bigramovým jazykovým modelem natrénovaným z korpusu, který obsahuje 3.5 GB textů. Trénovací korpus byl tvořen převážně novinovými články.

Nové texty jsou stahovány každý den a přidávány ke korpusu a nový jazykový model je přepočítáván z nového korpusu. K dispozici jsou také přepisy zpráv zpravodajství z aktuálního dne. Tyto přepisy jsou však přidány až následující den.

Žádné odhadování vah není prováděno. Tento postup má simulovat nasazení přepisovacího systému v praxi, kdy nových dat z aktuálního dne je velmi málo a

jejich dělení na held-out data a testovací data by ještě snížilo množství testovacích dat. V praxi většinou není čas na každodenní ladění vah. Z literatury [32, 33, 34, 35] je také zřejmé, že zlepšení úspěšnosti rozpoznávání lze očekávat je minimální. Výsledky přidávání jsou uvedeny v tabulce 9.1. Výsledky rozpoznávání pro 14 dní před a po datumu přepsání jsou uvedeny v příloze A v tabulce A.1.

Tabulka 9.1: Závislost úspěšnosti rozpoznávání zpráv konkrétního datumu na textech z jiných datumů.

Nahrávky z	7.12.2005	9.12.2005	12.12.2005
Přidané texty	úspěšnost rozpoznávání (Acc) %		
1.12.2005	72.78	75.75	76.43
2.12.2005	73.32	75.75	76.43
3.12.2005	73.35	75.72	76.39
4.12.2005	73.20	75.78	76.39
5.12.2005	73.53	75.75	76.43
6.12.2005	73.32	75.75	76.39
7.12.2005	**73.20**	75.84	76.43
8.12.2005	74.19	75.97	76.36
9.12.2005	74.89	**76.15**	76.58
10.12.2005	74.80	82.13	76.74
11.12.2005	75.19	82.13	76.71
12.12.2005	75.22	82.13	**76.74**
13.12.2005	75.31	82.13	82.41
14.12.2005	75.22	82.04	82.38

První sloupec reprezentuje zprávy, pro které nejsou k dispozici přepisy. Přepisy ke zprávám v posledních dvou sloupcích k dispozici jsou. Proto je zde vidět výrazný skok v úspěšnosti rozpoznávání ode dne, kdy jsou přidány přepisy zpráv, které mají být rozpoznány. Tím se do jazykového modelu dostaly přesně ty promluvy, které mají být rozpoznány. Zprávy mají být přepsány v den vysílání, proto jsou výsledky úspěšnosti rozpoznávání po datu vysílání v praxi nepotřebné. Výsledky pouze kvantifikují vliv přidání promluv, které mají být rozpoznány, do jazykového modelu. Výsledky také ukazují, že pokud se přidávají již přepsané zprávy z minulých dní, je jazykový model výrazněji lepší, než když tyto přepisy nejsou k dispozici.

V následujícím experimentu je přidávání přepisů zpráv eliminováno. Každodenní nová data jsou přidávána pouze z novinových článků. Výsledky jsou uvedeny tabulce 9.2. Výsledky rozpoznávání pro 14 dní před a po datumu přepsání jsou uvedeny v příloze A v tabulce A.2.

Přestože se přidání přepisu zpráv na úspěšnosti rozpoznávání v takto krátkém časovém úseku téměř neprojevilo, v delším časovém horizontu má přidávání

Tabulka 9.2: Závislost úspěšnosti rozpoznávání zpráv konkrétního datumu na textech z jiných datumů bez přidávání přepisů zpráv.

Nahrávky z	7.12.2005	9.12.2005	12.12.2005
Přidané texty	úspěšnost rozpoznávání (Acc) %		
1.12.2005	73.05	76.00	76.08
2.12.2005	73.14	76.00	76.11
3.12.2005	73.11	75.97	76.02
4.12.2005	73.17	75.97	76.14
5.12.2005	73.41	76.00	76.08
6.12.2005	73.20	76.00	76.11
7.12.2005	**73.38**	75.94	76.18
8.12.2005	73.32	75.97	76.05
9.12.2005	73.44	**76.00**	76.21
10.12.2005	73.50	76.33	76.33
11.12.2005	73.65	76.49	76.30
12.12.2005	73.83	76.49	**76.27**
13.12.2005	73.65	76.43	76.43
14.12.2005	73.89	76.61	76.27

známých přepisů zpráv do textového korpusu pozitivní dopad na úspěšnost rozpoznávání. Toto také indikuje první sloupec v tabulkách A.1 a A.2 zobrazující větší časový rozsah. Statisticky významné zlepšení bylo zaznamenáno pouze pro nahrávku ze 7.12.2005, jejíž přepisy nebyly k dispozici, p-hodnota = 4.0e-03. Z experimentů je patrné, že přidávání přepsaných zpráv pomáhá udržovat velmi kvalitní jazykový model. K udržování již kvalitního jazykového modelu tudíž není potřeba častých aktualizací.

9.3 Tématická adaptace jazykového modelu pro lékařský systém

Jazykový model pro lékařský diktovací systém vyvíjený v Laboratoři počítačového zpracování řeči Technické univerzity v Liberci [43] je tvořen unigramy, proto je adaptace slovníku a jazykového modelu velmi svázána. V sekci 5.3 je uveden postup čištění korpusu lékařských dokumentů použitých pro vytvoření slovníku jazykového modelu pro tento diktovací systém.

Slovník a jazykový model vytvořený z lékařského korpusu byl adaptován se slovníkem a jazykovým modelem původního diktovacího systému [42], protože původní slovník obsahuje:

- fonetický přepis speciálních znaků jako je . , ; atd.,
- slova pro ovládání diktování a jejich fonetický přepis, například vymaž slovo, vyber druhý,
- více všeobecných slov, která mohou být i v lékařských zprávách použita.

9.3.1 Spojování slovníků

Před adaptací bylo provedeno zarovnání hodnot absolutních četností výskytu slov tak, aby 1000 nejčetnějších slov v obou slovnících mělo přibližně stejnou četnost výskytu. U těchto slov byl zjištěn faktor udávající kolikrát je v průměru četnost n-tého slova z většího slovníku větší než četnost n-tého slova menšího lékařského slovníku. Tento faktor byl použit pro úpravu malého slovníku. Oba slovníky byly setříděny podle četnosti slov v příslušných korpusech.

Adaptace slovníku a jazykového modelu proběhla dvěma způsoby.

Přidání k malému, kdy byla přidána nejčetnější slova z původního diktovacího systému k novému lékařskému slovníku.

Přidání k velkému, kdy byly oba slovníky spojeny a ze spojení byla vybrána nejčetnější slova. Četnosti stejných slov byly sečteny.

Nakonec byla přidána všechna slova pro ovládání diktování.

9.3.2 Experimenty

Lékařský korpus obsahuje 1,5 milionů slov, z toho 68 tisíc různých. Textová data jsou z oblasti kardiologie. 20 tisíc nejčetnějších slov bylo vybráno z lékařského korpusu do lékařského slovníku. Původní diktovací systém obsahoval slovník s 400 tisíci slovy vybranými z korpusu zahrnujícího 300 milionů slov, z toho bylo 1.9 milionu slov různých. Původní korpus byl vytvořen převážně z novinových článků. Výsledný slovník obsahuje 40000 slov, což znatelněji zvýší rychlost rozpoznávání na pomalejších počítačích oproti slovníku s 400000 slovy.

Experimenty byly prováděny s rozpoznávačem izolované řeči vyvinutém v Laboratoři počítačového zpracování řeči Technické univerzity v Liberci. Testovací promluvy obsahovaly 2804 slov. Úspěšnost rozpoznávání na všeobecné řeči je 90 %.

Tabulka 9.3 ukazuje úspěšnost rozpoznávání diktování lékařských zpráv. Úspěšnost rozpoznávání je vyjádřena v procentech správně rozpoznaných slov ze všech testovacích slov. Třetí sloupec ukazuje úspěšnost rozpoznávání, pokud nezáleží na velikosti písmen rozpoznaného slova. Rozpoznávač nabízí i 5 dalších

nejpravděpodobnějších slov. Tyto alternativy lze rychle vybrat ze seznamu, čímž je zrychlena oprava chyb. Čtvrtý sloupec ukazuje úspěšnost rozpoznávání, pokud správně rozpoznané slovo bylo mezi 3 nejpravděpodobnějšími nabízenými alternativami. Poslední sloupec je obdoba předposledního, ale nezáleží na velikosti písmen rozpoznaného slova.

Tabulka 9.3: Úspěšnost rozpoznávání diktování lékařských zpráv. CI znamená, že nezáleží na velikosti písmen rozpoznaného slova.

Slovník	správně	správně CI	3 nejlepší	3 nejlepší CI	OOV
Původní	58 %	62 %	65 %	77 %	21 %
Přidání k malému	82 %	86 %	89 %	96 %	6 %
Přidání k velkému	79 %	83 %	88 %	97 %	6 %

9.3.3 Zhodnocení

Z experimentů je patrné, že adaptace slovníku, tedy přidání slov, která nejsou ve slovníku, má významný vliv na zvýšení úspěšnosti rozpoznávání. Adaptace jazykového modelu sečtením četnosti výskytů slov při metodě přidávání k velkému slovníku je nevhodná, neboť je tímto sečtením nový jazykový model znatelně poškozen. Tak veliké zvýšení úspěšnosti rozpoznávání bylo dosaženo proto, že lékařský slovník a jazykový model je velmi odlišný od slovníku a jazykového modelu původního diktovacího systému.

Kapitola 10

Úprava textového výstupu rozpoznávače

Jazykový model lze použít i v jiných částech systému rozpoznávání řeči než jen pro výběr nejpravděpodobnější promluvy ve Viterbiho algoritmu. Úprava výstupního textu rozpoznávače je vhodným kandidátem na použití jazykového modelu. Pokud jsou ve slovníku uvedena slova pouze malými písmeny z důvodu snížení velikosti slovníku a jazykového modelu a rychlejšího rozpoznávání, může být jiný jazykový model aplikován na dodatečný převod malých písmen na velká.

Další úprava výstupu rozpoznávače spočívá v přidávání interpunkce. Výstup rozpoznávače je pak mnohem čitelnější. Automatické vkládání interpunkce je uvedeno v následující sekci. Jazykový model může být také použit k čištění textu, ze kterého je vytvářen korpus a následně počítán jazykový model pro rozpoznávač. V tomto případě není vhodné použít ten samý jazykový model vytvořený pro účely rozpoznávání, neboť obsahuje jen omezenou množinu slov a je zbytečně veliký. Jazykový model je možné použít například přepisování zkratek do správného pádu. Při přepisování zkratek lze spíše použít jazykový model založený na třídách.

Ve výstupu rozpoznávače se číslovky objevují pouze expandované do slovní formy. Expanze, která pomohla při vytváření textového korpusu nyní snižuje čitelnost výstupu, neboť číslice v podobě písmen jsou nepřirozeně dlouhé.

10.1 Automatická interpunkce

Většina rozpoznávačů řeči produkuje sekvenci mezerami oddělených slov. Interpunkce vytváří výstup rozpoznávače čitelnější pro čtení. Interpunkce je také důležitá pro další zpracování textu, jako je získávání informací z rozpoznaného textu, strojový překlad, morfologická analýza, atd.

Automatická interpunkce se snaží najít konce vět a vložit do nich tečky a čárky v souvětí. K odhadnutí správné pozice interpunkce v češtině je třeba kombinovat informace z akustické části promluvy, jazykového modelu a morfologické analýzy. Detailní morfologická analýza je však závislá na znalosti pozic interpunkčních znamének. Morfologická analýza může být částečně nahrazena jazykovým modelem. V této práci je použit morfologický analyzátor Jana Hajiče [44], který známému slovu přiřadí jeho morfologické kategorie: slovní druh, osobu, číslo, pád, atd. Analyzátor přiřazuje slovu všechny možné kategorie. Pokud analyzátor slovo nezná, pak je označeno znakem „X“ na pozici slovního druhu.

Z literatury je patrné, že dosavadní systémy provádějící automatickou interpunkci kombinují znalost průběhu základní frekvence (F0), n-gramového jazykového modelu, délky fonémů [40] a případně i morfologických značek [41]. Průběh F0 je po částech linearizován a jsou z něj extrahovány různé příznaky, například sklon lineárních úseku. Článek [41] vychází z [40], je ale zaměřen na češtinu.

V práci [41] bylo pozorováno, že v češtině pozice čárek závisí spíše na informacích z jazykového modelu, zatímco pozice teček je spíše určena akustickou částí promluvy. Tentýž článek používá morfologický analyzátor k seskupení málo častých slov.

Automatická interpunkce je v této práci založena na automaticky nalezených produkčních pravidlech, která jsou naučena pro tečky a čárky zvlášť.

Rozpoznávač řeči používaný v této práci [2] je schopen rozpoznat také některé hluky [47] jako je ticho, nádech, atd. Informace o hlucích je použita místo akustické informace, čímž je umožněna automatická interpunkce výstupu rozpoznávače bez znovupoužití rozpoznávaného signálu. Pozorováním výstupů rozpoznávače bylo zjištěno, že hluky potřebnou akustickou informaci pro účely automatické interpunkce zachovávají.

Před tím, než mohou být naučena pravidla pro vkládání teček, je nutné zarovnat rozpoznané promluvy a referenční přepisy těchto promluv. Tím se tečky dostanou do výstupu rozpoznávače a je tak možné vytvořit trénovací a testovací data.

Pravidla pro vkládání čárek jsou odvozena z velkého textového korpusu, neboť tak je možné zajistit spolehlivější odhad jazykového modelu použitého pro tento účel.

Oba druhy pravidel je třeba následně spojit a ošetřit případy, kdy dochází k jejich současné aplikaci. Morfologický analyzátor je následně použit k odstraňování interpunkce, která neodděluje věty, neboť věty většinou obsahují podmět a přísudek nebo jeden z nich.

10.1.1 Automatické vkládání teček

Pravidla pro vkládání teček jsou odvozena z rozpoznaných šumů (ticho, nádech), které rozpoznávač vkládá do svého výstupu. Tyto šumy jsou označeny čísly 0 až 5 a pomlčkou [47]. Příklad nahrávky s rozpoznaným šumem je uveden na obrázku 10.1.

Obrázek 10.1: Nahrávka s rozpoznaným šumem

Výstup rozpoznávače:	...podle ní nerespektuje soukromí lidí **3** i ministr zahraničí ho vidí jako chybu
Vložená interpunkce:	...podle ní nerespektuje soukromí lidí**.** I ministr zahraničí ho vidí jako chybu.

Akustická data

Akustická data jsou tvořena zprávami 3 největších českých televizních stanic. V tabulce 10.1 jsou uvedeny podrobnější informace o použitých akustických datech. Jeden řečový segment obsahuje jednu nebo více vět. Segmenty jsou promluvy jednoho mluvčího.

Tabulka 10.1: Akustická data

	Trénovací	Testovací
Řečových segmentů	498	339
Délka promluv	81 min	53 min
Segmentů s tečkou	262	181
Segmentů s čárkou	377	251
Celkem teček	521	407
Celkem čárek	839	553

Akustická data jsou součástí databáze COST278 ze sekce 4.2.1.

Zarovnání přepisů

Abychom zahrnuli interpunkci do výstupu rozpoznávače, je nutné zarovnat referenční přepisy s interpunkcí bez označení šumů s přepisy generovanými rozpoznávačem, které obsahují šumy, ale neobsahují interpunkci. Zarovnání si musí poradit i s případy, kdy rozpoznávač udělal chybu.

Zarovnávání je prováděno metodou dynamického programování, stejně jako vyhodnocování výsledku rozpoznávače v kapitole 8.

Pokud rozpoznávač vloží do svého výstupu slovo navíc (inzerce) a interpunkce má být právě v těchto místech, pak může být interpunkce ve výstupu rozpoznávače vložena před či za toto slovo v závislosti na směru zarovnávání. Pokud je kolem inzerce nějaký rozpoznaný šum, pak by interpunkce měla být vložena na místo tohoto šumu. Takovéto automatické zarovnávání lze docílit různými cenami substitucí. V zarovnávání byla použita cena delece a inzerce rovná 7. Cena substituce dvou slov byla 10 a cena substituce šumu a slova byla 13.

Pravidla pro vkládání teček

Sekvence šumů indikující interpunkci je hledána gramatickou evolucí [60]. Tečka a čárka jsou v trénovací fázi chápány jako jediná interpunkce. V české větě lze občas nahradit čárku spojující věty tečkou bez ztráty smyslu.

Délka hledaných sekvencí šumu nebyla limitována. Cílem bylo najít takové sekvence, aby přesnost umístění teček byla maximální. Pokud je vložená tečka na místě zarovnaného rozpoznaného přepisu, pak je umístěna přesně. Ostatní umístění teček, či jejich vynechání je považováno za chybu.

Populace gramatické evoluce čítala 500 jedinců. Turnajová selekce byla použita pro výběr rodičů a steady state selekce pro vytváření nové populace. Diverzita populace byla udržována metodou LICE [62]. Učená pravidla jsou produkční pravidla následujícího formátu:

$$\textbf{pokud}\ (\text{sekvence šumů}),\ \textbf{pak}\ \text{napiš tečku místo sekvence} \tag{10.1}$$

Duplicitní tečky jsou odstraněny po aplikaci pravidel na celý výstup rozpoznávače.

10.1.2 Automatické vkládání čárek

Pravidla pro vkládání čárek jsou založena na jazykovém modelu a znalosti morfologických kategorií slov, které jsou zjištěny morfologickým analyzátorem [44]. Pravidla jsou odvozena z textového korpusu.

Textový korpus

Textový korpus je tvořen převážně novinovými články. Trénovací část korpusu obsahuje 360 milionů slov, z toho 2 miliony jsou různé. 90 % slov je oddělena mezerou, 7 % čárkou.

Korpus obsahuje 13.5 milionů čárek. 72 % čárek jsou následována spojkou, zájmenem, příslovcem nebo předložkou. Tyto slovní druhy jsou dále chápány jako obvyklá spojovací slova. V korpusu je obsaženo přibližně 3000 různých obvyklých spojovacích slov a 3 miliony ostatních slov, před kterými se vyskytuje čárka.

Testovací část korpusu je různá od trénovací, jsou použity jiné datumy vydání článků. Testovací data obsahují 18 milionů slov, z nichž 0.4 milionu je různých.

Žádná korekce interpunkce nebyla prováděna, proto mohou trénovací i testovací data obsahovat chyby.

Pravidla pro vkládání čárek

Pravidla pro vkládání čárek jsou automaticky odvozena z textového korpusu. Formát pravidel je:

$$\textbf{pokud}\ (\text{sekvence slov}),\ \textbf{pak}\ \text{napiš čárku před sekvenci} \tag{10.2}$$

Kvůli velkému množství různých slov a ještě většímu množství slovních spojení je kompletní prohledávání sekvencí slov téměř nemožné. Proto jsou sekvence slov omezeny pouze na obvyklá spojovací slova, spojky, zájmena, příslovce a předložky, což je založeno na pozorování korpusu. Slovní druh je určen morfologickým analyzátorem. Protože rozpoznávač pracuje s omezeným slovníkem, je možné slovní druhy určit jednou pro slova ve slovníku, jiná slova se ve výstupu rozpoznávače nemohou objevit.

Obvyklá spojovací slova jsou dosti nezávislá na svém delším okolí, proto jsou pro snížení výpočetní náročnosti prohledávány sekvence obsahující maximálně 2 slova.

Odvození pravidel pro vkládání čárek

Pravidlo pro vkládání čárek je aplikováno, pokud je splněna jeho podmínka, jinak je ponechán původní oddělovač slov, mezera. Tento přístup je založen a pozorování, že většina slov je oddělena mezerou.

Pravidla s jedním slovem v podmínce jsou odvozena jako první. Slova mohou být oddělena 3 různými separátory: mezera, čárka, jiný separátor. Maximálně věrohodné odhady následujících pravděpodobností jsou vypočteny z trénovacího korpusu.

$$P(\text{čárka}|\text{spojovací slovo}) \tag{10.3}$$
$$P(\text{mezera}|\text{spojovací slovo}) \tag{10.4}$$
$$P(\text{jiný separátor}|\text{spojovací slovo}) \tag{10.5}$$

Následně je určeno maximum z těchto pravděpodobností

$$\underset{i}{\operatorname{argmax}}(P(\text{separátor}_i|\text{spojovací slovo})) \tag{10.6}$$

Pokud je maximum (10.6) pro pravděpodobnost (10.3), je vytvořeno nové pravidlo.

Po odvození pravidel s jediným slovem v podmínce jsou odvozena pravidla se slovní dvojicí v podmínce, která upravují „zjemňují“ případy, kdy jsou pravidla s jedním slovem v podmínce příliš „hrubá“. Tedy pravděpodobnost (10.3) je sice největší, ale hodnota je ještě nízká a lze nalézt podstatné výjimky. Sekvence dvou spojovacích slov může způsobit, že se čárka píše jen před celou dvojici slov. Například v sekvenci „... případ, ve kterém ...“ je čárka až před slovem „ve“. Jindy může čárka úplně zmizet, i když se před oběma spojovacími slovy častěji čárka píše, pokud se vyskytují samostatně. Například před spojením „... a že ...“ se čárka ve většině případů nepíše.

Odvození pravidel se slovní dvojicí je obdobné jako odvození pravidel s jedním slovem, jen možností pro psaní interpunkce je více. Kvůli řídkému výskytu spojení, kdy jsou slova oddělena jinými separátory, než je čárka a mezera, jsou odhadovány jen pravděpodobnosti pro následující případy:

$$\text{mezera}\ (\text{první slovo})\ \text{mezera}\ (\text{druhé slovo}) \tag{10.7}$$

$$\text{čárka}\ (\text{první slovo})\ \text{mezera}\ (\text{druhé slovo}) \tag{10.8}$$

$$\text{mezera}\ (\text{první slovo})\ \text{čárka}\ (\text{druhé slovo}) \tag{10.9}$$

$$\text{čárka}\ (\text{první slovo})\ \text{čárka}\ (\text{druhé slovo}) \tag{10.10}$$

Případy (10.9) a (10.10) jsou podmnožiny pravidel s jedním slovem v podmínce.

Nová pravidla jsou vytvářena, pokud je maximum

$$\underset{i}{\operatorname{argmax}}(P(\text{případ}_i|\text{slovní dvojice})) \tag{10.11}$$

pro případy (10.7) nebo (10.8).

Při vytváření pravidel je také předpokládáno, že před slovy, která nejsou obvyklá spojovací slova, je častěji mezera než čárka. V případech, kdy je před jiným než obvyklým spojovacím slovem čárka je pro určení této čárky potřeba detailnější morfologická analýza, než jakou může bigramový model nabídnout. Proto je méně chyb provedeno, pokud je mezera před těmito slovy zachována. Tento předpoklad je založen na zkoumání textového korpusu.

Aplikace pravidel

Dva typy pravidel pro vkládání teček a čárek byly naučeny odděleně. Je proto nutné vyřešit konflikty, kdy může být aplikováno více pravidel najednou.

Jako první jsou aplikována pravidla vkládající tečky. Duplicitní tečky a zbylé šumy jsou odstraněny. Text již obsahuje jen tečky a slova.

Před aplikací pravidel pro vkládání čárek jsou tato pravidla upravena tak, aby akceptovala kromě mezery i tečku mezi slovy. Aby nedocházelo ke konfliktům

při aplikaci pravidel s jedním slovem v podmínce, která jsou všeobecnější, a speciálnějšími pravidly, se slovní dojicí v podmínce, jsou pravidla aplikována podle následujícího schématu:

- Pravidla se slovní dvojicí jsou aplikována jako první před pravidly s jedním slovem v podmínce.
- Pokud je aplikováno nějaké pravidlo, žádné jiné pravidlo nesmí být aplikováno na použitá slova. Další aplikace může začít až za použitými slovy.

Duplicitní interpunkce je následně odstraněna, přičemž je preferováno odstranění tečky před čárkou. Každý přepsaný řečový segment je nakonec ukončen tečkou.

Redukce interpunkce pomocí morfologického analyzátoru

Převážná většina vět oddělených interpunkcí obsahuje podmět nebo přísudek. Morfologický analyzátor je použit pro identifikaci podmětu a přísudku.

Přísudek je v češtině snadno identifikovatelný jako aktivní forma slovesa. Identifikovat podmět je mnohem složitější, neboť podmět má často stejnou tvar jako předmět. Nelze jen podle slova rozpoznat předmět od podmětu. Proto každé slovo, které může být podmět, je jako podmět chápáno, neboť není prováděna žádná detailnější morfologická analýza.

Text s vloženou interpunkcí je procházen zleva doprava, a pokud prošlý úsek neobsahuje podmět nebo přísudek a je oddělen interpunkcí, je tato interpunkce odstraněna.

10.1.3 Experimenty

Experimenty byly prováděny na testovací části akustických dat. Před vyhodnocováním byly rozpoznané promluvy zarovnány s referenčními přepisy. Následně byla provedena automatická interpunkce výstupu rozpoznávače. Výstupy automatické interpunkce a zarovnávání byly porovnávány.

Výsledky jsou uvedeny ve 4 mírách: úspěšnosti inzerce (Acc), precision (P), recall (R) a F-measure (F) definované [68]:

$$F = \frac{2RP}{R+P}. \tag{10.12}$$

V prvním experimentu nejsou aplikována žádná pravidla pro vkládání interpunkce, a proto slouží jako baseline. Výsledky jsou uvedeny v tabulce 10.2.

V následujícím experimentu jsou pravidla pro vkládání teček a čárek aplikována odděleně vždy na čistý výstup rozpoznávače. Tabulka 10.3 ukazuje výsledky pro tento případ.

Tabulka 10.2: Žádná interpunkce není vložena, baseline

Úspěšnost (Acc)	Precision (P)	Recall (R)	F-measure (F)
88.27 %	100.00 %	88.27 %	93.77 %

Tabulka 10.3: Tečky a čárky jsou aplikovány samostatně

Typ pravidel	Úspěšnost (Acc)	Precision (P)	Recall (R)	F-measure (F)
Pouze tečky	90.10 %	75.06 %	90.10 %	81.90 %
Pouze čárky	90.35 %	77.17 %	90.35 %	83.24 %

V dalším experimentu je zkoumán vliv odstraňování interpunkce pomocí morfologického analyzátoru. Výsledky jsou uvedeny v tabulce 10.4.

Tabulka 10.4: Vliv odstraňování interpunkce pomocí morfologického analyzátoru

Odstranění interpunkce	Úspěšnost (Acc)	Precision (P)	Recall (R)	F-measure (F)
Ne	92.05%	77.05%	92.05%	83.88%
Ano	91.91%	80.22%	91.91%	85.67%

V posledním experimentu byla provedena kompletní automatická interpunkce. Při vyhodnocování tohoto experimentu nebyl rozdíl mezi typem interpunkce, tečky i čárky byly chápány jako jediné interpunkční znaménko. Výsledky jsou uvedeny v tabulce 10.5

Tabulka 10.5: Kompletní automatická interpunkce, přičemž při vyhodnocování byly tečky i čárky chápány jako jedno interpunkční znaménko

Úspěšnost (Acc)	Precision (P)	Recall (R)	F-measure (F)
92.81%	88.04%	92.81%	90.36%

10.1.4 Zhodnocení

Uvedené experimenty ukazují významné zlepšení oproti ponechání výstupu rozpoznávače jako mezerami oddělený proud slov. Byla dosažena úspěšnost vkládání interpunkce 92.81%. Čitelnost výstupu rozpoznávače může být dále vylepšena kapitalizací písmen po tečkách.

Vyhodnocování nesprávně rozpoznané promluvy je dosti problematické, neboť chyba může i nemusí ovlivnit správnou pozici interpunkce. Jedno špatně rozpoznané slovo může změnit význam celé věty a snížit úspěšnost pravidel založených na jazykovém modelu.

Pravidla pro vkládání teček byla natrénována pouze na nahrávkách televizních zpráv, proto je jejich úspěšnost na tomto typu akustických dat závislá. Není ovšem problém po získání jiných nahrávek pravidla pro vkládání teček přetrénovat.

Kapitola 11

Závěr

Tvorba lingvistické vrstvy systému automatického rozpoznávání mluvené češtiny je v této práci pojata jako komplexní problém. V průběhu práce bylo vytvořeno množství programů, postupů a vylepšení umožňující automatizaci adaptace slovníku a jazykového modelu.

V díle jsou diskutovány problémy týkající se různých zdrojů textových dat, jejich získávání a čištění. Byly vytvořeny robustní autonomní programy schopné získávat data 24 hodin denně 356 dní v roce. Zároveň byly uvedeny postupy normalizace textu jak pro všeobecné texty z novinových zpráv, tak i pro speciální lékařské texty, kterých je většinou málo a obsahují mnoho chyb a cizích slov. Jsou uvedeny metody identifikace cizích slov umožňující aplikaci správných fonologických pravidel. Byl experimentálně prokázán pozitivní vliv různých normalizací na úspěšnost rozpoznávání. Během práce byl zvětšen textový korpus o více než 100 %.

Další část díla je zaměřena na slovník a fonetickou transkripci. Je uvedena závislost pokrytí českého textového korpusu na velikosti slovníku. Dále je uvedena vlastní metoda vylepšení fonetické transkripce spočívající v natrénování nových fonologických pravidel, která jsou následně přidána k existujícím fonologickým pravidlům. Nová pravidla jsou natrénována pomocí Gramatické evoluce. Výhodou uvedené metody je, že neobjevuje již známá pravidla. Nová naučená pravidla jsou ihned připravena k aplikaci. Poslední část kapitoly týkající se slovníku se zabývá přidáváním slovních spojení do slovníku. Jde o jednoduchý a téměř bezpracný způsob zvýšení úspěšnosti rozpoznávání. Slovní spojení jsou vybírána na základě vhodné míry. Vhodnost různých měr je experimentálně ověřena. Úspěšnost rozpoznávání byla touto metodou zvýšena z 74.48 % na 77.94 %.

V kapitole zabývající se jazykovým modelem jsou diskutovány otázky efektivní implementace výpočtu jazykového modelu s velkým slovníkem tak, aby jej bylo možné spočítat na běžně dostupných počítačích v přijatelném čase. Jsou uvedeny vlastní implementace výrazně zrychlující výpočet jazykového modelu oproti

dosavadnímu programu používaném v Laboratoři počítačového zpracování řeči. Kapitola 7 uvádí výsledky experimentů zjišťujících vliv velikosti slovníku a interpunkce na úspěšnost rozpoznávání. Je též uveden průběh nalézání nových bigramů při výpočtu jazykového modelu, ze kterého je patrné, že pro slovník obsahující 312 tisíc slov stále existuje množství bigramů, jejichž hodnota je odhadnuta nepřesně pro nedostatek dat. K přesnějšímu odhadu málo četných bigramů je však potřeba velké množství dat. Je proto nutné sbírat další texty do textového korpusu.

Další kapitola se zabývá detailní analýzou výsledků rozpoznávání. Abychom mohli efektivně zlepšovat rozpoznávač, je nutné vědět, které chyby jsou při rozpoznávání nejčetnější. V kapitole je uvedena vlastní modifikace běžně používané metody vyhodnocování výsledků. Pomocí uvedené modifikace je možné přesněji určit, která slova jsou rozpoznávačem vložena, vypuštěna, či zaměněna za jiné. Jsou zde též uvedeny a kvantizovány nejčetnější chyby rozpoznávačů a chyby vzniklé díky příčestí minulému a chyby psaní „y“ a „i“.

V kapitole zabývající se adaptací jazykového modelu jsou provedeny experimenty týkající se tématické a časové adaptace jazykového modelu. Především experimentů ukazujících vliv přidávání nových textů na úspěšnost rozpoznávání je v literatuře velmi málo. Ve většině publikací je uveden pouze vliv adaptace na perplexitu která, jak se v literatuře ukazuje, má malý vztah ke skutečné úspěšnosti rozpoznávání. Z provedených experimentů vyplývá, že k údržbě kvalitního jazykového modelu není třeba častých aktualizací. Občasné přidání aktuálních dat se pozitivně projeví na úspěšnosti rozpoznávání. Je též zřejmá nutnost přidávání nových slov do slovníku, aby tato slova mohla být rozpoznávána.

Poslední kapitola se zabývá úpravou textového výstupu z rozpoznávače s cílem zvýšit čitelnost tohoto výstupu. V kapitole je uvedena vlastní modifikace existujících metod automatického vkládání interpunkce. Publikovaná metoda je oproti ostatním metodám schopna odvodit pozice interpunkce pouze z výstupu rozpoznávače, a to díky informacím o různých šumech, které výstup rozpoznávače obsahuje. Byla dosažena 92.81% úspěšnost automatické interpunkce.

Většina vytvořených programů je aktivně používána jak v Laboratoři počítačového zpracování řeči, tak je i součástí komplexních komerčních produktů Laboratoře počítačového zpracování řeči.

Další rozvoj lingvistické vrstvy systému automatického rozpoznávání mluvené češtiny spočívá v neustálém sběru nových dat potřebných pro zlepšování odhadu n-gramů a získávání nově se objevujících slov a slovních spojení. Je možné též provádět sofistikovanější normalizace textového korpusu. Analýza výsledků rozpoznávače je neustále diskutovaným tématem zejména díky tomu, že dosavadní vyhodnocovací metody nereflektují vliv chyby na sémantickou a pragmatickou část jazyka. Automatická interpunkce je založena na informaci obsažené v jazykovém modelu a šumech z výstupu rozpoznávače. Dále je možné důkladněji analyzovat i signál vstupující do rozpoznávače a zakomponovat nově získané

informace.

Literatura

[1] Jan Nouza, Tomáš Nouza, and Petr Červa. A multi-functional voice-control aid for disabled persons. In *Proceedings of the SPECOM 2005*, Patras, Greece, 2005.

[2] Jan Nouza, Jindřich Žďánský, Petr David, Petr Červa, Jan Kolorenč, and Dana Nejedlová. Fully automated system for czech spoken broadcast transcription with very large (300k+) lexicon. In *Proceedings of the Interspeech 2005*, Lisbon, Portugal, 2005.

[3] Akinobu Lee, Tatsuya Kawahara, and Kiyohiro Shikano. Julius — an open source real-time large vocabulary recognition engine. In *Proceedings of the European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1691–1694, 2001.

[4] SJ Young. The htk hidden markov model toolkit: Design and philosophy. Technical Report 153, Department of Engineering, Cambridge University (UK), 1993.

[5] Xuedong Huang and Alex Acero Hsiao-Wuen Hon. *Spoken Language Processing: a guide to theory, algorithm, and system development*. Prentice Hall PTR, Upper Saddle River, New Jersey 07458, 2001. ISBN 0-13-022616-5.

[6] Jan Nouza, editor. *Počítačové zpracování řeči - cíle, problémy, metody*. Technická univerzita v Liberci, 1 edition, 2001. 55-087-01.

[7] Liang Gu, Jayanth Nayak, and Kenneth Rose. Discriminative training of tied-mixture hmm by deterministic annealing.

[8] M. Kurimo. Using self-organizing maps and learning vector quantization for mixture density hidden markov models. *Acta Polytechnica Scandinavica, Mathematics Computing and Management in Engineering Series*, 87:1–55, 1997.

[9] Zdena Pálková. *Fonetika a fonologie češtiny*. Karolinum, Praha, 2 edition, 1997.

[10] Jan Kolorenč. Evolving phonological rules using grammatical evolution. In *Proceedings of the 8th International Student Conference on Electrical Engineering–POSTER 2004*, Prague, 5 2004. [CD-ROM].

[11] Sean M. Burke. *Perl & LWP*. O'Reilly, 2002. ISBN 0-596-00178-9.

[12] Jan Nouza, Dana Nejedlova, Jindrich Zdansky, and Jan Kolorenc. Very large vocabulary speech recognition system for automatic transcription of czech broadcast programs. In *Proceedings of the ICSLP 2004*, October 2004.

[13] Roeland Ordelman, Arjan van Hessen, and Franciska de Jong. Compound decomposition in dutch large vocabulary speech recognition. In *Eurospeech 2003*, september 2003.

[14] Andre Breton, Pablo Fetter, and Peter Regel-Brietzmann. Compound words in large-vocabulary german speech recognition systems. In *Fourth International Conference on Spoken Language Processing (ICSLP 96)*, October 1996.

[15] Mikko Kurimo, Antti Puurula, Ebru Arisoy, Vesa Siivola, Teemu Hirsimaki, Janne Pylkkonen, Tanel Alumae, and Murat Saraclar. Unlimited vocabulary speech recognition for agglutinative languages. In *Human Language Technology, Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2006*, New York, USA, June 5-7 2006.

[16] Mathias Creutz and Krista Lagus. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 workshop on Morphological and phonological learning*, pages 21–30, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[17] Pavel Ircing and Josef Psutka. Two-pass recognition of czech speech using adaptive vocabulary. In *TSD 2001, Lecture Notes in Artificial Intelligence 2166*, pages 273–277, Berlin, Heidelberg, 2001. Springer-Verlag.

[18] George Saon and Mukund Padmanabhan. Data-driven approach to designing compound words for continuous speech recognition. *IEEE transactions on speech and audio processing*, 9(4):327–332, 2001. ISSN 1063-6676.

[19] Jan Kolorenč, Jan Nouza, and Petr Červa. Multi-words in the tv/radio news transcription system. In *Speech and Computer International Conference - Specom 2006*, pages 103–106, Petersburg, Russia, June 2006. ISBN 5-7452-0074-x.

[20] International Phonetic Association. Report on the 1989 kiel convention. *Journal of the Phonetic Association*, 19(12), 1989.

[21] Jan Nouza, Josef Psutka, and Jan Uhlíř. Phonetic alphabet for speech recognition of czech. *Radio Engineering*, 6(4):16–20, December 1997.

[22] Marek Volejník. Fonetická transkripce psané a mluvené češtiny pro účely automatického zpracování řeči. Master's thesis, Technická univerzita v Liberci, Fakulta mechatroniky a mezioborových inženýrských studií, 1999.

[23] Johnson and Mark. A discovery procedure for certain phonological rules. In *Proceedings of the Tenth International Conference on Computational Linguistic*, pages 334–347. Stanford, 1984.

[24] Rilley and D. Michael. A statistical model for generating pronounciation networks. In *Proceedings of the IEEE ICASSP-91*, pages 737–740, 1991.

[25] Terrence J. Sejnovski and Charles R. Rosenberg. Parallel networks that learn to read aloud. In *Cognitive Science*, volume 1598, pages 179–211, 1986.

[26] Daniel Gildea and Daniel Jurafsky. Automatic induction of finite state transducers for simple phonological rules. In *Proceedings of the Meeting of the Association for Computational Linguistics*, pages 9–15, 1995.

[27] Jindra Drábková. Punctuation effect on classed-based language model for czech language. In *Proceedings of the Electronic Speech Signal Processing 2005, ESSP 2005*, pages 267–272, Prague, Czech Republic, Semtember 2005. ISBN 3-938863-17-X.

[28] Jan Hajič. Morphological tagging: data vs. dictionaries. In *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics*, pages 94–101, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[29] Ian H. Witten and Timothy C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4):1085–1094, 1991.

[30] Andreas Stolcke. Srilm – an extensible language modeling toolkit. In *International Conference on Spoken Language Processing (ICSLP)*, 2002.

[31] Philip Clarkson and Ronald Rosenfeld. Statistical language modeling using the CMU–cambridge toolkit. In *Proc. Eurospeech '97*, pages 2707–2710, Rhodes, Greece, 1997.

[32] Kristie Seymore and Ronald Rosenfeld. Large-scale topic detection and language model adaptation. Technical Report CMU-CS-97-152, Computer Science Department, Carnegie Mellon University, June 1997.

[33] David Janiszek, Frederic Bechet, and Renato de Mori. Integrating map and linear transformation for language model adaptation. In *Proceedings of the 6th International Conference on Spoken Language Processing, ICSLP2000*, volume 2, pages 895–898, Beijing, October 2000.

[34] Simo Broman and Mikko Kurimo. Methods for combining language models in speech recognition. In *Proceedings of the Interspeech 2005*, pages 1317–1320, Lisbon, Portugal, 2005.

[35] Javier Dieguez-Tirado, Carmen Garcia-Mateo, and Antonio Cardenal-Lopez. Effective topic-tree based language model adaptation. In *Proceedings of the Interspeech 2005*, pages 1317–1320, Lisbon, Portugal, 2005.

[36] Kristie Seymore and Ronald Rosenfeld. Using story topics for language model adaptation. In *Proceedings of Eurospeech '97*, pages 1987–1990, Rhodes, Greece, 1997.

[37] Dietrich Klakow. Log-linear interpolation of language models. In *Proceedings of the International Conference on Spoken Language Processing*, Sydney, Australia, 1998.

[38] Michiel Bacchiani and Brian Roark. Unsupervised language model adaptation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 224–227, 2003.

[39] Stanley Chen, Kristie Seymore, and Ronald Rosenfeld. Topic adaptation for language modeling using unnormalized exponential models. In *Proceedings of the ICASSP '98*, 1998.

[40] Elizabeth Shriberg, Andreas Stolcke, Dilek Hakkani-Tür, and Gökhan Tür. Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communication*, 32(1-2):127–154, 2000.

[41] Jáchym Kolář, Josef Psutka, and Jan Švec. Automatic punctuation annotation in czech broadcast news speech. In *Proceedins of 9-th International Conference Speech and Computer (SPECOM 2004)*, St. Petersburg, Russia, 2004.

[42] Jan Nouza and Tomáš Nouza. A voice dictation system for a million-word czech vocabulary. In *Proceedings of the of ICCCT 2004*, ISBN 980-6560-17-5, pages 149–152, Austin, USA, 8 2004.

[43] Jan Kolorenč and Tomáš Klimovič. Cardiology language model for voice dictation. In *Proceedings of the 14th Czech-German Workshop*, pages 93–97, Prague, September 2004. ISBN 80-86269-11-6.

[44] Jan Hajič, Eva Hajičová, Petr Pajas, Jarmila Panevová, Petr Sgall, and Barbora Vidova Hladká. Prague dependency treebank. CDROM LDC2001T10, Linguistic Data Consortium, University of Pennsylvania, 2001.

[45] Jindřich Žďánský. Detection of acoustic change-points in audio streams and signal segmentation. *Radioengineering*, 14(1):37–40, 2005.

[46] Petr Červa and Jan Nouza. Supervised and unsupervised speaker adaptation in large vocabulary continuous speech recognition of czech. In *Text, Speech and Dialogue (TSD)*. Springer-Verlag, Heidelberg, 2005.

[47] Jindřich Žďánský and Martin Kroul. Semi-automatic non-speech events database formation. In *Proceedings of the 8th International Student Conference on Electrical Engineering - POSTER 2004*, May 2004.

[48] Jan Kolorenč. Automatic punctuation of automatically recognized speech. In *Proceedings of the Electronic Speech Signal Processing 2005*, Prague, Czech Republic, September 2005. ISBN 3-938863-17-X.

[49] An Vandecatseye, Jean-Pierre Martens, Joao Neto, Hugo Meinedo, Carmen Garcia-Mateo, Javier Dieguez, France Mihelic, Janez Zibert, Jan Nouza, Petr David, Matus Pleva, Anton Cizmar, Harris Papageorgiou, and Christina Alexandris. The cost278 pan-european broadcast news database. In *Proceedings of the LREC2004: Fourth international conference on language resources and evaluation*, Lisbon (Portugal), 2004.

[50] NIST. Matched pairs sentence-segment word error (mapsswe) test. online⟨http://www.nist.gov/speech/tests/sigtests/mapsswe.htm⟩.

[51] Larry Gillick and Stephen Cox. Some statistical issues in the comparison of speech recognition algorithms. In *ICASSP 89*, pages 532–535, 1989.

[52] Dana Nejedlová, Jindra Drábková, Jan Kolorenč, and Jan Nouza. Lexical, phonetic, and grammatical aspects of very-large-vocabulary continuous speech recognition of czech language. In *Proceedings of the Electronic Speech Signal Processing 2005*, pages 224–231, Prague, Czech Republic, September 2005. ISBN 3-938863-17-X.

[53] Kamil Chalupníček. Rozpoznávání diktované řeči pro medicínské aplikace. Master's thesis, Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2004.

[54] Martin Vokurka and Jan Hug. *Velký lékařský slovník*. Maxdorf, 4 edition, 8 2004. online ⟨http://www.maxdorf.cz/maxdorf/ls.html⟩.

[55] Jan Kábrt and Jan Kábrt. *Lexicon medicum*. Avicenum Praha, 1 edition, 1988.

[56] Hornická Zaměstnanecká Pojišťovna. Vše o lécích. online ⟨http://www.hzp.cz/leky/⟩, 8 2004.

[57] Jean-Luc Gauvain, Lori Lamel, Gilles Adda, and Mich'ele Jardino. The limsi 1998 hub-4e transcription system. In *Proceedings of the DARPA Broadcast News Workshop*, Herndon, VA, 1999.

[58] Gerhard Backfried and Roser Jaquemot Caldes. Spanish broadcast news transcription. In *Proceedings of the EUROSPEECH-2003*, pages 1561–1564, 2003.

[59] Dana Nejedlová. Fonetická transkripce češtiny pomocí třívrstvé neuronové sítě. Technical report, Technická univerzita v Liberci, Laboratoř zpracování řeči, Liberec, 200.

[60] Conor Ryan, J. J. Collins, and Michael O' Neill. Grammatical evolution: Evolving programs for an arbitrary language. In *Proceedings of the First European Workshop on Genetic Programming*, volume 1391, pages 83–95, Paris, 14-15 1998. Springer-Verlag.

[61] Vladimír Mařík, Olga Štěpánková, Jiří Lažanský, and kolektiv. *Umělá inteligence 3*. Academia. ISBN 8020004726, EAN 9788020004727.

[62] Jan Kolorenč. Získávání znalostí z dat pomocí gramatické evoluce. Master's thesis, České vysoké učení technické v Praze, Fakulta elektrotechnická, 2004.

[63] Wayne Ward, Holly Krech, Xiuyang Yu, Keith Herold, George Figgs, Ayako Ikeno, Dan Jurafsky, and William Byrne. Lexicon adaptation for lvcsr: speaker idiosyncracies, non-native speakers, and pronunciation choice. In *Proceedings of the Pronunciation Modeling and Lexicon Adaptation for Spoken Language Technology (PMLA)*, pages 83–88, 2002.

[64] Michael Finke and Alex Waibel. Speaking mode dependent pronunciation modeling in large vocabulary conversational speech recognition. In *Proceedings of the Eurospeech '97*, pages 2379–2382, Rhodes, Greece, 1997.

[65] Michael Riley, William Byrne, Michael Finke, Sanjeev Khudanpur, Andrej Ljolje, John McDonough, Harriet Nock, Murat Saraclar, Charles Wooters, and George Zavaliagkos. Stochastic pronunciation modelling from handlabelled phonetic corpora. In *Proceedings of the ETRW on Modeling Pronunciation Variation for Automatic Speech Recognition , Kerkrade, Netherlands*, pages 109–116, 1998.

[66] Andre Berton, Pablo Fetter, and Peter Regel-Brietzmann. Compound words in large-vocabulary german speech recognition systems. In *Proceedings of the ICSLP 96*, 1996.

[67] Roeland Ordelman, Arjan van Hessen, and Franciska de Jong. Compound decomposition in dutch large vocabulary speech recognition. In *Proceedings of the Eurospeech 2003*, September 2003.

[68] Information retrieval. Wikipedia. online ⟨http://en.wikipedia.org/wiki/Information_retrieval⟩.

Dodatek A

Časová adaptace jazykového modelu

Tabulka A.1: Závislost úspěšnosti rozpoznávání zpráv konkrétního datumu na textech z jiných datumů.

Nahrávky z	7.12.2005	9.12.2005	12.12.2005
Přidané texty	úspěšnost rozpoznávání (Acc) %		
20051123	73.05	77.10	76.68
20051124	72.96	75.97	75.89
20051125	73.26	75.90	75.96
20051126	73.05	75.97	76.18
20051127	73.14	75.97	76.11
20051128	72.96	75.97	76.30
20051129	72.99	75.90	76.36
20051130	73.08	75.84	76.36
20051201	72.78	75.75	76.43
20051202	73.32	75.75	76.43
20051203	73.35	75.72	76.39
20051204	73.20	75.78	76.39
20051205	73.53	75.75	76.43
20051206	73.32	75.75	76.39
20051207	**73.20**	75.84	76.43
20051208	74.19	75.97	76.36
20051209	74.89	**76.15**	76.58
20051210	74.80	82.13	76.74
20051211	75.19	82.13	76.71
20051212	75.22	82.13	**76.74**
20051213	75.31	82.13	82.41
20051214	75.22	82.04	82.38
20051215	75.28	82.00	82.45
20051216	75.31	82.07	82.45
20051217	75.13	82.07	82.48
20051218	75.22	82.19	82.38
20051219	75.07	82.19	82.41
20051220	75.07	82.19	82.41
20051221	75.07	82.10	82.48
20051222	75.16	82.10	82.51
20051223	75.16	82.04	82.54
20051225	75.07	82.10	82.54
20051226	75.25	82.07	82.54
20051227	75.16	81.97	82.48
20051228	75.34	81.94	82.51

Tabulka A.2: Závislost úspěšnosti rozpoznávání zpráv konkrétního datumu na textech z jiných datumů bez přidávání přepisů zpráv.

Nahrávky z	7.12.2005	9.12.2005	12.12.2005
Přidané texty	úspěšnost rozpoznávání (Acc) %		
20051123	72.87	75.97	75.80
20051124	72.78	75.97	75.83
20051125	72.87	75.90	75.86
20051126	72.72	75.81	76.05
20051127	73.02	75.84	76.02
20051128	72.75	75.94	75.99
20051129	72.90	75.84	76.08
20051130	72.99	75.90	76.08
20051201	73.05	76.00	76.08
20051202	73.14	76.00	76.11
20051203	73.11	75.97	76.02
20051204	73.17	75.97	76.14
20051205	73.41	76.00	76.08
20051206	73.20	76.00	76.11
20051207	**73.38**	75.94	76.18
20051208	73.32	75.97	76.05
20051209	73.44	**76.00**	76.21
20051210	73.50	76.33	76.33
20051211	73.65	76.49	76.30
20051212	73.83	76.49	**76.27**
20051213	73.65	76.43	76.43
20051214	73.89	76.61	76.27
20051215	73.71	76.67	76.49
20051216	73.89	76.67	76.43
20051217	73.80	76.61	76.49
20051218	73.83	76.64	76.52
20051219	73.77	76.70	76.46
20051220	73.95	76.70	76.55
20051221	73.80	76.67	76.58
20051222	73.62	76.58	76.61
20051223	74.07	76.55	76.58
20051225	73.95	76.55	76.65
20051226	73.83	76.67	76.65
20051227	73.95	76.58	76.55
20051228	73.92	76.61	76.58

Dodatek B

Výsledky přidávání slovních párů do slovníku

Tabulka B.1: Slovní spojení přidávaná do slovníku.

Přidaných spojení	úspěšnost rozpoznávání (Acc)		
	PMI	četnost výskytu	četnost výskytu s předložkou na 1. místě
01000	74.59	75.40	75.37
02000	74.55	75.73	76.15
03000	74.55	75.95	76.11
04000	74.50	76.20	76.11
05000	74.64	76.05	76.52
06000	74.70	75.89	76.59
07000	74.67	75.78	76.81
08000	74.68	76.04	76.78
09000	74.64	76.28	76.89
10000	74.60	76.33	76.99
15000	74.63	76.82	77.68
20000	74.79	77.13	77.57
25000	74.99	77.37	77.65
30000	74.95	77.43	77.77
35000	74.92	77.57	77.88
40000	74.92	77.43	77.90
45000	74.94	77.69	77.94
50000	74.96	77.46	77.91
55000	75.02	77.45	77.90